THE UNIVERSITY of EDINBURGH

Predictability effects in language acquisition

John K Pate

Doctor of Philosophy

Institute for Language, Cognition and Computation
School of Informatics
University of Edinburgh

2013

Abstract

Human language has two fundamental requirements: it must allow competent speakers to exchange messages efficiently, and it must be readily learned by children. Recent work has examined effects of language predictability on language production, with many researchers arguing that so-called "predictability effects" function towards the efficiency requirement. Specifically, recent work has found that talkers tend to reduce linguistic forms that are more probable more heavily. This dissertation proposes the "Predictability Bootstrapping Hypothesis" that predictability effects also make language more learnable. There is a great deal of evidence that the adult grammars have substantial statistical components. Since predictability effects result in heavier reduction for more probable words and hidden structure, they provide infants with direct cues to the statistical components of the grammars they are trying to learn.

The corpus studies and computational modeling experiments in this dissertation show that predictability effects could be a substantial source of information to language-learning infants, focusing on the potential utility of phonetic reduction in terms of word duration for syntax acquisition. First, corpora of spontaneous adult-directed and child-directed speech (ADS and CDS, respectively) are compared to verify that predictability effects actually exist in CDS. While revealing some differences, mixed effects regressions on those corpora indicate that predictability effects in CDS are largely similar (in kind and magnitude) to predictability effects in ADS. This result indicates that predictability effects are available to infants, however useful they may be.

Second, this dissertation builds probabilistic, unsupervised, and lexicalized models for learning about syntax from words and durational cues. One series of models is based on Hidden Markov Models and learns shallow constituency structure, while the other series is based on the Dependency Model with Valence and learns dependency structure. These models are then used to measure how useful durational cues are for syntax acquisition, and to what extent their utility in this task can be attributed to effects of syntactic predictability on word duration. As part of this investigation, these models are also used to explore the venerable "Prosodic Bootstrapping Hypothesis" that prosodic structure, which is cued in part by word duration, may be useful for syntax acquisition. The empirical evaluations of these models provide evidence that effects of syntactic predictability on word duration are easier to discover and exploit than effects of prosodic structure, and that even gold-standard annotations of prosodic structure provide at most a relatively small improvement in parsing performance over

raw word duration.

Taken together, this work indicates that predictability effects provide useful information about syntax to infants, showing that the Predictability Bootstrapping Hypothesis for syntax acquisition is computationally plausible and motivating future behavioural investigation. Additionally, as talkers consider the probability of many different aspects of linguistic structure when reducing according to predictability effects, this result also motivates investigation of Predictability Bootstrapping of other aspects of linguistic knowledge.

Acknowledgements

I would first like to acknowledge my supervisor Sharon Goldwater for her support, insights, and (always constructive!) criticism over the course of this dissertation. Her advice has improved my thinking, my writing, and, many times over, my presentation skills. I would also like to acknowledge my secondary supervisor Simon King and my SICSA supervisor Robin Lickley; their valuable feedback has punctuated milestones in my time at Edinburgh.

Several members of the Linguistics and Informatics community have also been influential during my time at Edinburgh. Alice Turk has always been eager to talk about prosody, predictability effects, and ambiguity. Moreno Coco taught me almost everything I know about mixed effects modeling, and everything I know about model selection. Nathaniel Smith arrived just in time to challenge me to analyze my dependency parser properly, leading me to one of the more unexpected but more interesting results. Trevor Fountain was the source of not only technical insight, introducing me to `git`, but also companionship during American college football games. I'm also grateful to the attendees of the ProbModels, ML-for-NLP, DevLing and DevPhon reading groups for stress-free overviews of material I would not have understood on my own, and crucial feedback on presentations and abstracts.

The faculty, staff, and students of the Ohio State language community during my time there, prior to coming to Edinburgh, played a formative role in the genesis and development of my interest in language. In particular, I would like to acknowledge Detmar Meurers for kindling my initial interest in computational linguistics with the boundless energy of his introductory parsing course (and also seeing me through my first conference paper), and Chris Brew for developing a sense of the wealth of information available from a statistical view. Additionally, I am indebted to Cynthia Clopper who introduced me to the notion of predictability effects and gave me hours of interaction with real speech (by annotating it). Laura Wagner also played a crucial role in developing my interest in language acquisition, giving me a place in her lab and, more importantly, many hours of off-the-cuff discussion about language development. My interest in computational methods as applied to language acquisition were first grounded through conversations with Anton Rytting, who shared his data with me, spent hours talking with me as an undergraduate, and first pointed out Sharon Goldwater as a potential supervisor. Finally, the attendees and presenters at Phonies and Clippers enriched my initial development many times over, with fascinating presenta-

tions and discussions from the aforementioned along with Eric Fosler-Lussier, Mary Beckman, Jeff Holliday, Kathleen Hall, and many others that together constituted a beautiful and entrancing whirlwind introduction to language.

My friends have also been a source of encouragement, humility, and inspiration. Chris, Thom, Teager, Raymond, Peggy Kittila, Jennifer Yi, Brent Biglin, John Petrus, thank you for showing that the more things change, the more they stay the same. Nicholas Petricca, because of you I can say I've performed and partied with a rock star.

I would also like to thank my incredible fiancée Lin Shen for her support, love, and ever-gentle encouragement ("GRADUATE! NOW!") throughout our time in Scotland, and also for expanding my world to include another continent.

Finally, I must acknowledge my family, who have more than anything else made me who I am. My sisters Liz and Abby endured my many attempts to lecture them on things I half-understood with, if not quite patience, not outright hostility. And the ultimate acknowledgement goes to my parents, Kenton and Cathy Pate, who have taught me about learning, life, and love.

This work was carried out with funding from the Scottish Informatics and Computer Science Alliance, who also provided generous funds for travel to a conference and a summer school, and also funding from the Scottish Overseas Research Studentship Award Scheme.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*John K Pate*)

Preface

Much of the material in this dissertation has been published.

- Some of the material in Chapter 4 was published in Pate and Goldwater (2011a).
- The material in Chapter 5 was published in Pate and Goldwater (2011b).
- Some of the material in Chapter 6 was published in Pate and Goldwater (2013).

Table of Contents

Chapter 1

Introduction

Human language has two fundamental requirements: it must allow competent speakers to exchange messages efficiently, and it must be readily learned by children. Recent work has examined effects of language predictability on language production, with many researchers arguing that so-called "predictability effects" function towards the efficiency requirement. Specifically, recent work has found that talkers tend to reduce linguistic forms that are more probable, such as frequent words, frequent syntactic constructions, or expressions that refer to entities that are already in the common ground, more heavily. Although the concrete experiments in this dissertation will focus on phonetic reduction in terms of word duration (which is relatively easy to measure and correlates with other kinds of reduction), this dissertation takes a broad view of reduction, interpreting any (subconscious) choice that makes the linguistic signal shorter or less distinct without changing the meaning as reduction. For example, saying a word more quickly, deleting a phone from a word, undershooting a vowel, or omitting an optional word entirely are all various kinds of reduction. The focus of this dissertation is the proposal and evaluation of the "Predictability Bootstrapping Hypothesis" that predictability effects make language easier to learn.

To explore the possiblity that predictability effects function towards learnability, we will first examine the prevalence of predictability effects in child-directed speech (CDS). Previous work has established that predictability effects are common in adult-directed speech (ADS), but, other than the work presented in this dissertation, it has not been established that predictability effects exist in CDS. To quantify the existence and extent of predictability effects in CDS, this dissertation examines spontaneous speech corpora. Several statistical techniques are combined to mitigate complicated confounds inherent in spontaneous speech, including mixed effects regression (to con-

trol for hidden confounds such as shared talkers), model selection (to find a model that is both tractable and well-controlled), and residualization (to isolate correlations among control variables from variables of interest).

The corpus work concludes, in short, that predictability effects in CDS are largely the same as those in ADS. This result provides good evidence that predictability effects are readily available for children to exploit, however useful they may be. A secondary result, that predictability effects in CDS are not identical to predictability effects in ADS, addresses the hypothesis that predictability effects are about making language communication more efficient. Research so far has yielded little direct evidence that predictability effects have an appreciable effect on communication. Without such direct evidence, it is possible that predictability effects have a negligible effect on communicative efficiency and are primarily just a side-effect of how brains make sentences. However, by showing that talkers modulate predictability effects according to listener characteristics and the communicative environment, we provide an important new sort of evidence that predictability effects are related to communication.

Having established that predictability effects are available to children, we will examine how useful they might be for learning about syntax in an unsupervised setting. Specifically, we will incorporate word duration measures into unsupervised grammar induction systems, producing systems that implement "durational bootstrapping," use these systems to see if word duration can improve parsing performance, and assess the extent to which these systems recover the specific relationship between word duration and syntax that is involved in Predictability Bootstrapping. A great deal of work over the last few decades has shown that some notion of probability is represented in the adult grammars infants are trying to learn. Since predictability effects reduce a word when it is in a low probability structure, they provide infants with direct, observable evidence about the probability of the unobserved syntactic structure under the grammar they are trying to learn. Thus, if the "durational bootstrapping" models recover a relationship in which shorter words are associated with more certain structures, they will indicate that syntactic predictability effects are evident in the data. If the models moreover produce more accurate parses, then they will indicate that predictability effects are practically useful. We will call the hypothesis that linguistic reduction is useful for language acquisition by way of predictability effects the "Predictability Bootstrapping Hypothesis." We will see that very simple representations of word duration lead to improvements in the performance of two very different types of grammar induction systems, and moreover we will find evidence that the driving factor in this improve-

ment is indeed an association of shorter words with more certain structures.

The Predictability Bootstrapping Hypothesis is a version of what Morgan and Demuth (1996) called a "Phonological Bootstrapping Hypothesis" because it proposes that children use suprasegmental cues to gain other kinds of linguistic information. We will also compare the Predictability Bootstrapping Hypothesis with another more established Phonological Bootstrapping Hypothesis called the "Prosodic Bootstrapping Hypothesis." This hypothesis relies on the observation that prosody, the organizational structure of speech, often produces prosodic groupings of words whose boundaries coincide with the boundaries of syntactic constituents. Prosodic Bootstrapping proposes that the prosodic groupings provide a good enough initial cue to constituency structure to get children started with syntax acquisition. If children can recover prosodic structure from the acoustic signal reliably enough, and if prosodic structure corresponds closely enough with syntactic structure, then prosody should provide a good initial cue about syntax for infants. Our modeling experiments will indicate that prosodic structure is useful for syntax acquisition, if the assumed grammar form is expressive enough, but that effects of syntactic predictability are easier to exploit.

This dissertation makes both theoretical and empirical contributions. The primary theoretical contribution is the Predictability Bootstrapping Hypothesis, which points out how variation in linguistic redundancy could provide infants with hitherto unappreciated observable evidence for the statistical grammar they are trying to learn. In formulating explicit computational models for Prosodic Bootstrapping, this dissertation provides a secondary theoretical contribution in highlighting previously unacknowledged computational difficulties inherent in the Prosodic Bootstrapping of syntactic structure.

The primary empirical contribution of this dissertation is a computational demonstration that Predictability Bootstrapping is plausible for the case of using word duration to learn about syntax, using two different modeling approaches and grammar formalisms. On the way to this result, this dissertation also shows, for the first time, that predictability effects exist in child-directed speech. Additionally, the modeling experiments will suggest that prosodic phrasing does interact with syntactic structure in a way that is useful for unsupervised learning, but not necessarily by *coinciding* with syntactic structure. Finally, as a secondary empirical result, this dissertation also shows that talkers adjust predictability effects in response to listener and channel characteristics, bolstering the proposal that predictability effects can be understood, at least in part, in functional terms as an adaptation for communicative efficiency.

The rest of the dissertation is organized as follows. Chapter 2 provides more detail on attested predictability effects and why they might improve communicative efficiency, including a short tutorial on the Noisy Channel Theorem. Chapter 3 discusses bootstrapping accounts in language acquisition, presents a dimensionality-reduction view under which Predictability Bootstrapping should be easy, and outlines the computational modeling philosophy for evaluating bootstrapping accounts in this dissertation. Chapter 4 presents a series of regressions on spontaneous child-directed and adult-directed speech corpora, providing evidence predictability effects are actually present in speech to children (and, secondarily, providing evidence that the speech types have slightly different predictability effects). Chapter 5 presents a series of HMM-based unsupervised chunkers, and shows that learning from words and prosodic annotation outperforms learning from words alone, and that learning from words and word duration measures further outperforms learning from words and prosodic annotation. Chapter 6 presents a series of unsupervised dependency parsers, based on the Dependency Model with Valence (DMV), which show that word duration information can provide a substantial benefit to dependency parsing as well, and that most of this benefit appears to be due to predictability effects. Chapter 7 summarizes the contributions of this dissertation, and discusses possible future directions.

Chapter 2

Information Theory and Predictability Effects

2.1 Introduction

In recent years, it has been suggested that predictability effects are a mechanism for optimal communication in an information-theoretic sense. This chapter makes this proposed relation explicit. In Section 2.2, predictability effects are discussd in the context of traditional linguistic theory, and various kinds of attested predictability effects are presented. Section 2.3 introduces the efficiency-adaptation view of predictability effects. Section 2.4 reviews information theory, focusing on its relevance to predictability effects. Section 2.5 points out some remaining open issues with interpreting predictability effects as optimization in an information-theoretic sense.

2.2 Predictability Effects

Predictability effects are, in the broad terminology of Saussure, the tendency for talkers to reduce the *signifier* when the *signifique* is highly probable. This dissertation focuses on phonetic reduction, and more specifically on phonetic reduction in terms of word duration. Predictability effects, however, have been found for many sorts of *signifier*/*signifique* pairs. A canonical example lies in the lexicon itself: frequent words tend to have short phonological encodings, while infrequent words tend to have long phonological encodings. Such predictability effects are rooted in slow, diachronic processes which reduce a phonological *signifier* when it signifies a frequent lexical *signifique*. It has also been found (e.g. Frank and Jaeger, 2008; Jaeger, 2010; Ferreira

and Dell, 2000) that talkers will omit optional syntactic function words, such as the complementizer *that* and the infinitive marker *to*, precisely in more probable contexts. Such predictability effects are presumably rooted in syntactic processes, with a lexical *signifier* elided, essentially completely reduced, when it encodes a highly probable syntactic *signifique*.

Most work, however, has focused on phonetic reduction. Zhao and Jurafsky (2009) found, for example, that lexical tones in Cantonese are more dispersed and have a higher $f0$ for low-frequency words. Priva (2008) showed that consonants will be deleted if they are highly probable across words with similar phonological forms (and so provide little disambiguating information about those similar words). Aylett and Turk (2006) showed that vowels tend to be more heavily centralized in terms of first and second formants when the are pronounced as part of highly probable words. Finally, the bulk of work on phonetic reduction has focused on word duration, presumably because it is relatively easy to measure and correlates heavily with other measures. Bard et al. (2000) showed that words will be pronounced more quickly when they have already been mentioned in the discourse. Gahl and Garnsey (2004) and Gahl et al. (2006) showed, in laboratory studies, that verbs and verb dependents tend to be pronounced more quickly when they are in a highly-probable syntactic frame. Tily et al. (2009) found the same tendency in a treebank of spontaneous speech (these results will be of paramount importance when exploring the learnability requirement). Pluymaekersa et al. (2005) found that Dutch words tended to be pronounced more quickly, and have more deleted segments, when they had high unigram frequency, or were highly probable given the preceding or following word. Aylett and Turk (2004) and Bell et al. (2009) showed, on different spontaneous speech corpora, that words tend to be pronounced more quickly when they are highly probable in terms of unigram frequency, conditional probability given preceding or following bigram, and when they are not first mentions.

2.3 Why we have Predictability Effects

It has been fairly well established that predictability effects do, in fact, exist, and exist on many levels of linguistic description. The next natural question is: why do predictability effects exist? In this dissertation, three mutually-compatible explanations are considered:

1. Accident: Predictability effects are a curiosity of how brains make sentences, but don't necessarily have any particular effects on communicative efficiency or language learnability.

2. Efficiency: However they arose historically or biologically, predictability effects function to make language a more efficient communication system in the sense of the Noisy Channel theorem (Shannon, 1948).

3. Learnability: However they arose historically or biologically, and regardless of their effect on the communication of messages between adults, predictability effects make language more learnable by providing infants with observable evidence about hidden structure.

The first possibility would be best handled in a dissertation about human evolution, and in this dissertation is treated only as a null hypothesis: predictability effects are *only* a curiosity of how brains make sentences. For example, neural network models of lexical retrieval (e.g. TRACE McClelland and Elman, 1986) usually rely on some kind of competition in the activation of different lexical entries; when a node corresponding to a lexical entry is sufficiently activated to inhibit other lexical nodes, the word is recalled. These networks can be designed such that frequently-accessed nodes have a higher resting activation or inhibit competitors more strongly, ultimately causing frequently-accessed nodes to be accessed more quickly. If this is how lexical retrieval works, then we would observe lexical predictability effects that do not necessarily have much to do with improving efficiency. Chapter 4 discusses this null hypothesis in greater detail. The third possibility will be discussed and evaluated in Chapters 3, 5, and 6. We examine the second possibility in Chapter 4 and now.

2.3.1 Predictability effects as optimization for communication

There is a long tradition in linguistic theory of hypothesizing that predictability effects function towards efficient communication. Zipf (1949) proposed the "Principle of Least Effort," that talkers seek to minimize their average effort over time, as an explanation for why frequent words tend to be shorter. This principle did not provide a rigorous framework for measuring this inverse correlation, relying instead on the appealing intuition that effort is minimized by making the frequent case the short case.

While Zipf pointed out a relationship between word *type* form and frequency, Lindblom (1990) introduced an efficiency-based argument on a word *token* basis. Lindblom

(1990) sought to explain why vowels are sometimes pronounced very distinctly, with precise tongue movements often very far from the center of the mouth, and why vowels are sometimes pronounced very indistinctly, with ambiguous tongue movements near the center of the mouth. To do this, Lindblom (1990) first pointed out that the distinct pronunciations require greater energy (because articulators are moving farther and exerting greater force to maintain precision) but produce an acoustic signal that more readily differentiates a given vowel, while indistinct pronunciations require less energy but produce a more ambiguous acoustic signal. Lindblom (1990)'s "**H**yper- and **H**ypo-articulation" theory (H&H thory) proposes that talkers seek to use the more distinct pronunciation only when listeners will have difficulty successfully recognizing the speech. H&H theory does not specify how talkers would come to know when listeners might have difficulty; in particular, unlike Zipf's "Principle of Least Effort," H&H theory does not propose that word probabilities would be a determining factor. Nevertheless, since H&H theory posits that more distinct, more effortful linguistic forms occur when the listener is less certain about the message, it is an attempt to explain predictability effects as an efficiency optimization.

Aylett and Turk (2004), with expansions in Aylett and Turk (2006) and Turk (2010), presents the "Smooth Signal Redundancy" hypothesis, which aims to ground efficiency-based explanations for predictability effects in information theory. The Smooth Signal Redundancy hypothesis begins by pointing out that the non-reduced form in various predictability effects can be viewed as a form which provides redundant information. In the case of Lindblom's vowels, for example, it is often enough to know a vowel's height *or* its backness for lexical identification: providing clear cues to both is redundant. Aylett and Turk (2004) examines an even clearer case of redundant information: word duration. A word that is pronounced slowly provides approximately the same acoustic information at each point in the word for more milliseconds. The Smooth Signal Redundancy hypothesis proceeds to point out that a communicative system which provides more redundancy for less probable elements is more efficient in an information-theoretic sense (e.g. Shannon, 1948). Exactly how and why this kind of redundancy is efficient will be discussed shortly in Section 2.4

To evaluate the Smooth Signal Redundancy hypothesis, Aylett and Turk look at syllable durations on the HCRC Map Task (Anderson et al., 1991) and look for correlations between a word's predictability and the duration of its syllables. Aylett and Turk measure a word's probability in three different ways. First, they count the number of times a word has appeared in a dialogue before the token under consideration. This

measure is supposed to correspond to a discourse-level notion of givenness, such that entities which have already been introduced to the common ground are more likely to be mentioned again. Second, they look at the overall frequency of the word type (using a different corpus). Third, they look at the probability of a word given the preceding bigram (using probabilities from yet another larger corpus). Aylett and Turk (2004) report longer syllable durations for words that are more predictable under each measure. This result provides evidence that phonetic redundancy is modulated by word probability, for a variety of notions of "word probability," on a token-by-token basis. Aylett and Turk (2006) establishes (on a different, more heavily-controlled corpus of professionally-read speech) the same basic relationship between various notions of word probability and vowel distinctiveness. Turk (2010) proposes that prosodic phrasing, in the sense that will be discussed in Section 3.2.3.2, is also manipulated for efficiency in an information-theoretic sense (although data that bears directly on this point is not examined).

Frank and Jaeger (2008), with expansion in Jaeger (2010), presents largely the same hypothesis under the name "Uniform Information Density" (UID), and evaluates it with respect to morphosyntactic reduction rather than phonetic reduction. To evaluate the UID hypothesis, Frank and Jaeger (2008) examine, on a portion of the Switchboard dataset of spontaneous telephone dialogues (Bresnan et al., 2002), when talkers use contractions (e.g. "we're," "don't") and when talkers do not (e.g. "we are", "do not"). Clearly, contracted forms are shorter, contain fewer distinguishing features, and are just generally less redundant than full forms. Frank and Jaeger extract several instances of the words "be," "have," and "not," which are in syntactic environments that allow contraction, and see if talkers choose the contracted form more often if the instance is more probable. Frank and Jaeger (2008) explore many different notions of instance probability, and find that, by and large, more probable instances are more likely to be contracted. This result provides evidence that morphosyntactic reduction works towards efficiency in an information-theoretic sense.

So there is a lot of research activity directed towards evaluating the general hypothesis that predictability effects help make language efficient, and both the Smooth Signal Redundancy hypothesis and the Uniform Information Density hypothesis frame this approach in information-theoretic terms. Chapter 3 will introduce the hypothesis that predictability effects help make language learnable. However, as our results also speak to the efficiency hypothesis, we will cover why predictability effects would lead to more efficient communication in an information-theoretic sense. In particu-

lar, while both the Smooth Signal Redundancy hypothesis and the UID hypothesis appeal to information-theoretic principles, and in particular the Noisy Channel Theorem of Shannon (1948), relevant details about information theory are usually omitted. Accordingly, the next section presents the Noisy Channel Theorem as it relates to predictability effects.

2.4 The Noisy Channel Theorem: A Tutorial

Shannon (1948) is a foundational paper that establishes many aspects of information theory in addition to the Noisy Channel theorem. In this section, we examine only what is necessary for a thorough understanding of the Noisy Channel Theorem, as it relates to natural language. In particular, the proofs of various theorems will not be presented; it is recommended that the interested reader consult either Shannon (1948) directly or the first eleven chapters of Mackay (2003).

Shannon (1948) formalizes communication as the endeavor by a sender to communicate a message (the Saussurean *signifique*) to a receiver. By hypothesis, the message cannot be *directly* sent to the receiver, so the sender encodes the message into a form, called the signal (the Saussurean *signifier*), that can be sent to the receiver. For example, the message might be a sequence of letters, but our available equipment is capable of transmitting only binary digits. Or, the message might be a lexical entry, but our equipment is capable of producing only cochleograms. For an initial example, suppose that our messages are sequences of 32 characters: A-Z, space, and the five punctuation symbols period, question mark, exclamation mark, comma, and apostrophe. Suppose also that our signal consists of sequences of binary digits zero and one. Let M represent the set of message characters, and S represent the set of signal characters.

A natural question to ask is: which sequence of signal characters should we use to encode each message character? Suppose the message characters are equiprobable and independent. One possible coding scheme would be to impose an ordering on all the message characters (perhaps following the POSIX standard), and encode each character with a series of '1's corresponding to its position in the ordering, using 0 to delimit message characters. So, the message "ABC" would be encoded "101101110." Now, is this a very good encoding? If we want to communicate quickly, we clearly want to use short codes rather than long codes. For this code, on average, we expect to use $\sum_{n=2}^{33} n\frac{1}{32} = \frac{560}{32} = 17.5$ signal characters to communicate one message character.

Shannon (1948) showed that the theoretical most efficient code will use exactly

$H = -\sum_{m \in M} P(m) \log_b(P(m))$ signal characters per message character, where the base b of the logarithm is the number of possible signal characters. H (with H pronounced 'eta,' not 'aytch') is the *entropy* of the source encoding the message. Intuitively, it represents how uncertain we are about the message. Concretely, this comes down to requiring longer signals, under the optimal code, when we are more uncertain (that is, the message is less probable).

Let's compute the number of signal characters used per message character by the optimal code. If it is close to 17.5 signal characters per message character, then the coding scheme defined above is close to optimal. Since we have 2 signal characters, the base of the logarithm is 2, and since all of our message characters are by hypothesis equiprobable, the term inside the summation is equal to $\frac{1}{32}\log_2(\frac{1}{32})$ for all message characters m. Multiplying this term by 32 instead of summing (as we have 32 message characters) and factoring the $\frac{1}{32}$ out, the expression reduces to $-32 \cdot \frac{\log_2(\frac{1}{32})}{32} = -\log_2(2^{-5}) = 5$ signal characters per message character. Evidently, the code defined above is not very good, since it uses three and a half times as many signal characters per message character as necessary. In this case, finding the most efficient code is straightforward: there are $2^5 = 32$ possible five-character signal sequences, which conveniently corresponds to the number of message characters, and so we can map these one-to-one with the message characters in whatever manner we like. Since this code is guaranteed to produce exactly five signal characters per message character, and the entropy of the source is exactly five signal characters per message character, it is not possible to obtain a shorter code than this.

Of course, messages of interest, in particular those suitable for modeling natural language, do not consist of sequences of independent, equiprobable characters. What kinds of codes are efficient for skewed probability distributions? The answer ends up being those codes which give longer encodings for low-probability messages, and shorter encodings for high-probability messages. There really are two points here: first, we should move from providing a code for each message character to providing a code for messages (that is, sequences of message characters), and, second, this code should provide longer encodings for low-probability messages.

The first point can be understood by modifying the above scenario only slightly: instead of having 32 message characters, we will have 22. The optimal code will now use $-22 \cdot \frac{1}{22}\log_2(\frac{1}{22}) = \log_2(22) \approx 4.46$ signal characters per message character. We could, in principle, keep using the same encoding used above, but it is guaranteed to be less efficient than the optimal code, since it uses five signal characters rather than 4.46

per message character. We can approach the optimal code by modeling sequences of characters. There are $22^2 = 484$ possible message sequences of length two. Since we have $2^9 = 512$ possible signal sequences of length 9, we can encode all 484 two-length sequences using 9 signal characters. This produces a code which uses $\frac{9}{2} = 4.5$ signal characters per message character, which is very close to the optimal rate of about 4.46 computed above. The optimal code can be more closely approximated by using longer chunks, and Theorem 3 of Shannon (1948) (which is not the Noisy Channel Theorem) shows that the optimal code can be approximated arbitrarily closely by increasing the chunk length, and Theorem 6 (still not the Noisy Channel Theorem) shows that this is true even if message characters are correlated rather than independent.

The second point, that lower-probability messages should receive longer encodings than higher-probability messages, is best understood by considering why the formula for the entropy of the optimal code takes the form $-\sum_{m\in M} P(m)\log_b(P(m))$. This looks suspiciously like an expected value computation. Indeed, when computing the average signal characters used per message character for our inefficient code above, we performed exactly an expected value computation, summing the length of each code, weighted by its probability. A bit of algebra shows:

$$H = -\sum_{m\in M} P(m)\log_b(P(m)) = \sum_{m\in M} P(m)\log_b\left(\frac{1}{P(m)}\right)$$

We can see from this that entropy just *is* an expected value computation, where the optimal number of signal characters for a message character m is the log of the inverse probability of m (this optimal number of signal characters for a particular m is called the *Shannon information* of m). In the course of proving Theorem 9, the noise**less** channel theorem, Shannon (1948) presents two different methods for constructing arbitrarily optimal codes. Devising a code that approximates codewords whose length comes out to the Shannon information for each message character is called *source coding*. A code is *redundant* to the extent that its codewords are longer than their Shannon information (together, this means that source coding in computer science typically tries to pick codes with no redundancy).

This alone is enough to argue that frequency effects on relatively static linguistic representations make the linguistic system more efficient in an information-theoretic sense. More frequent words tend to be shorter, for example, meaning that the phonological specification for a word is inversely proportional to its log frequency. The proportionality corresponds to freedom in choosing the base of the logarithm, due to

the following logarithmic identity:

$$\frac{\log_a(x)}{\log_a(b)} = \log_b(x)$$

Since the base of the logarithm corresponds to the cardinality of the coding system, this could correspond to a decision to use one set of phonological features over a different set. In principle, under the assumption that humans are nearly optimal, it would be possible to recover the cardinality of the phonological feature set by seeing which logarithm base leads most closely to equality between a word's citation form and its negative log probability. However, we know that the lexicon is not perfectly optimal in this sense, because phonological codes still have redundancy in, for example, phonotactics. We will shortly see that redundancy is useful for combating noise, but always leads to a less efficient code in the absence of noise.

Before turning to the Noiseless and Noisy channels, let's introduce and clarify some terminology. Since a signal in one alphabet can be straightforwardly converted to an equivalent signal in a different alphabet, it is customary to work through the math using whatever logarithmic base is convenient. Since I am examining this from a computational perspective, I will stick with a logarithmic base of 2. This means that all of the information theoretic measures are in **bi**nary digi**ts**, or bits. I have avoided this terminology so far to make explicit that bits of entropy correspond directly to average signal length, and to clarify whether I am talking about signal characters or message characters. In the discussion that follows, I will occasionally refer to bits, but do not forget that bits are just a unit of measurement for signal length.

Now, we will turn to the Noiseless and Noisy channels. First, we will consider the Noiseless channel (given in theorem 9 of Shannon (1948)), which introduces many important concepts. Specifically, a noiseless channel is defined to be a channel which always transmits to the receiver exactly the same signal that was sent out by the sender. Now, a channel will have limits so that it is capable of conveying only so many signal symbols per second. This limit is called the channel capacity C, and is measured in bits per second. In the context of natural language, several different factors could influence channel capacity. If we take our signal to consist of phonological feature specifications, the channel capacity will be limited by such factors as how quickly our articulators can move to produce a particular feature specification and the speed of sound.

The Noiseless Channel Theorem shows that, for a channel with capacity C and for signal source with entropy H, the maximum rate of transmission is just $\frac{C}{H}$. This is an intuitively appealing result, because the unit of H is bits per message symbol;

it expresses how much signal we must devote to each portion of the message. The units of C are just bits per second: how much signal we can transmit in a second. The bits cancel out in the dimensional analysis, and we end up with message symbols per second.

However, the Noiseless Channel Theorem is unrealistic in that the receiver always gets exactly the signal that the sender meant to transmit. In a natural language setting, this means that it assumes that there are no slips of the tongue, no sudden loud sounds that mask speech sounds, no background noise, and so on. The Noisy Channel Theorem relaxes this assumption by introducing random noise in the channel: some bits in the signal will be flipped at random. Clearly, all of the coding schemes examined so far will give rise to errors. Suppose we picked a five-bit code for the 32 characters, for example, that maps 00000 to 'A', 00001 to 'B', 00010 to 'C', and so on. One bit flip leads us to decode an entirely different message character. If our channel flips only 5% of bits, we still expect one error every four message characters: an effective error rate of 25%!

Shannon (1948) introduces three terms to clarify the ramifications of a noisy channel. First, rather than talking about just the entropy H, we talk about the entropy of the source signal $H(X)$. We also discuss the entropy of the received signal $H(Y)$. In the case of the noiseless channel, we have $X = Y$ and so $H(X) = H(Y)$. Finally, we have the entropy of the source signal *given* the received signal: $H(X|Y)$. Formally, this is just the entropy of the conditional probability distribution:

$$\begin{aligned} H(X|Y) &= \sum_{y \in Y} P(Y=y) H(X|Y=y) \\ &= -\sum_{y \in Y, x \in X} P(Y=y) P(X=x|Y=y) \log(P(X=x|Y=y)) \\ &= -\sum_{y \in Y, x \in X} P(Y=y, X=x) \log(P(X=x|Y=y)) \end{aligned} \tag{2.1}$$

Intuitively, this corresponds to how uncertain the receiver is about what the sender *meant* to send, after seeing the message that was received through the noisy channel. Since this is an entropy, it can be expressed in bits, and so, concretely, it corresponds to how many extra signal characters the receiver requires to be certain about what the sender meant to send. In a noiseless channel, this quantity is 0.

So, how might we go about providing those extra bits? The trick is to do something that makes no sense in a noiseless channel: encode the same information multiple

times. This introduction of *redundancy* to the signal is called *channel coding* (remember that the process of eliminating redundancy is called source coding). Pronouncing a word more slowly increases redundancy because it provides more copies[1] of the same cochleogram (although it may also allow articulations closer to the target).

To see why encoding the same information multiple times leads to a signal that is robust to noise, consider our five-bit coding system above. It produces, on average, one error every four characters in a channel with 5% noise. A simple way to make it resistant to noise is to repeat each binary digit three times, so 010101 becomes 000111000111000111. Clearly, this is suboptimal in a noiseless channel, since the new code is three times as long with no benefit. However, if one bit in a triple is flipped in a noisy channel, the receiver can still recover the intended signal by taking a majority vote for each triple. While a communication error can still occur if two or three bits are flipped in the same triple, this event is much less likely. Specifically, there are three possible ways to flip two bits in the same triple, and only one way to flip three bits. Since our noise is randomly distributed at 5% and independent, each two-bit flip occurs with probability $0.05^2 \cdot 0.95 = 0.002375$, and the three-bit flip occurs with probability $0.05^3 = 0.000125$. We thus expect an error error about the intended bit $3 \cdot 0.002375 + 0.000125 = 0.725\%$ of the time in our redundant code, down from 5% of the time in the original five-bit-per-message-character code. This leads to an expected error roughly every 25 message characters, an effective error rate of about 4% down from 25%.

Intuitively, this seems like a pretty good trade-off: a three-fold increase in signal length leads to a more than five-fold reduction in error rate. What if this error rate is still too high? One option would be to repeat each binary digit five times instead of just three, leading to a further diminished error rate at the expense of an even longer code. However, we ideally want codes that are both robust to noise and short. Is there some limit to how much longer we need to make our codes in order to achieve arbitrarily low error rates? It seems *prima facie* possible that signal lengths would have to go to infinity to drive error rates arbitrarily close to zero.

Recall that the Noise**less** Channel Theorem establishes that the minimum number of signal characters per message character is just the entropy of the source $H(X)$. We can think of this as the rate of information transfer for a noiseless channel if we use the optimal code. The Noisy Channel Theorem will establish a related limit for informa-

[1] Technically, since time is a continuous variable, there are infinitely many copies of infitesimal duration, and a slow pronunciation provides similar densities over cochleograms for a longer time span. All of the intuitions developed here generalize to the continuous case.

tion transmission across a noisy channel with arbitrarily low error rates.

As a starting point for thinking about how to combat noise, Theorem 10 (immediately before the Noisy Channel Theorem), proposes the following set-up: the sender and the receiver set up two channels, a main channel and a correction channel. The main channel uses a code without redundancy and so uses $H(X)$ bits per message character. The correction channel contains a device that sees the transmitted message at both the sender and the receiver, and transmits correction data to the receiver when the received signal does not match the sent signal. Theorem 10 proves that such a correction channel can correct all but an arbitrarily small number of errors as long as it is able to transmit at a rate greater than or equal to $H(X|Y)$. Intuitively, this is because, while $H(X)$ expresses the number of message characters per signal character, $H(X|Y)$ expresses the number of *possible sent signal characters* (aside from the true sent signal character) per *received* signal character. So $H(X|Y)$ is the average number of corrective, redundant signal characters we must transmit to clarify which signal is the intended signal.

The Noisy Channel theorem (theorem 11) itself simply proves that this extra information can be anticipated, in expectation, if we know what the noise is like, and provided ahead of time *without seeing the received signal*. Under such a set-up, we have a source that generates signals at a rate close to $H(X)$, and then we use our knowledge about the noise to add in the extra redundant information. While the two kinds of encoding can be performed in a single step (and are in the proof of theorem 11), a redundant code can be viewed as having two encoding schemes for two different probability distributions. The system for encoding messages into signals depends on the probability distribution over messages (source coding), and the system for encoding received signals into sent signals depends on our probability distribution over received signals given the sent signal, which in turn depends on the kind of noise (channel coding). The overall rate of such a system is then the entropy of the message minus the amount of redundant encoding needed:

$$R = H(X) - H(X|Y) \tag{2.2}$$

This is just a descriptive statement that applies to any redundancy scheme, not just an optimal or nearly optimal redundancy scheme, and is also called the *mutual information* of X and Y.

There are several related interpretations we can take for this equation. First, we can view it as saying that the rate of *correct* information transfer is the difference between

the *attempted* rate of information transfer (our $H(X)$ term) and the average number of errors that result. This interpretation is most natural if we are considering a code with little or no redundancy: it attempts a very small number of signal characters per message character, but, if there is significant noise, results in a large expected number of errors.

We can also view Equation 2.2 as saying that our source is capable of sending $H(X)$ information, and that it reserves $H(X|Y)$ of that information rate for encoding signal characters *instead of* message characters. The actual rate of communication of the message is then the difference between what the source is capable of encoding and how much of that capability is devoted to signal redundancy.

The rate of communication is $H(X) - H(X|Y)$, and the $H(X)$ term is influenced only by the probability distribution over messages, so anticipating and adapting to noise changes only the $H(X|Y)$ term. This means that the best code is the one which maximizes the difference $H(X) - H(X|Y)$ by adding exactly the right amount of redundancy to deal with expected errors due to noise (and the rate of this code, $\max(H(X) - H(X|Y))$, is the capacity of the noisy channel). So, we can have arbitrarily small error rates without having to resort to infinitely long codes.

So, how close to optimal is our triplet encoding scheme? Unfortunately, because the best code depends on the specific properties of the noise, computing the capacity of a noisy channel is much more difficult than computing the rate of a noiseless channel. Consider the triplet encoding scheme used above. If the source of noise flipped the same overall proportion of signal bits, but always flipped three consecutive bits rather than just one, our majority-vote encoding would be much less effective at combating noise. We'd need to adopt some encoding with redundancy that lasted longer than the noise corruption, which could involve encoding chunks of message characters rather than just one at a time (especially if the noise source is complicated). So the redundant encoding must be adapted to the kind of noise involved.

We can, however, understand what the optimal code should look like by examining Equation 2.1 in more detail. First, let's note that the conditional entropy term also looks like an expectation.

$$H(X|Y) = \sum_{y \in Y} P(Y = y) H(X|Y = y) \tag{2.3}$$

Following the intuitions above, this equation expresses how uncertain the receiver is after seeing the received message, and so is an expectation about the expected error

rate. To achieve a low error rate, then, we want $H(X|Y)$ to be small. Due to the rearrangement inequality, the right hand side of Equation 2.3 is minimized when small $P(Y = y)$ is associated with large $H(X|Y = y)$, and large $P(Y = y)$ is associated with small $H(X|Y = y)$. When $H(X|Y = y)$ is large, there is a lot of uncertainty for the receiver about this particular message y, and we are more likely to end up with an error. We can minimize the impact of large $H(X|Y = y)$ by ensuring that this particular y is rare: $P(Y = y)$ should be small. So a short and robust code is a code which only provides larger $H(X|Y = y)$ for less probable received signals. Now, $P(Y = y)$ can be small because it is either caused by a low-probability source signal $X = x$, or it has been subjected to an unusually high degree of noise, or both. In an artificial system, there will not be especially low-probability source signals, because the source code will have been defined according to the probability distribution over messages. Language shows a tendency towards this kind of source coding, because, e.g., frequent words tend to have shorter phonological forms, but it is not perfect, because, e.g., phonological forms still exhibit redundancies like phonotactics. Together, this means that a short and robust code provides longer redundant encodings for less probable *sent* signals, plus some extra length in the codes to account for the possibility of unusually severe noise.

2.4.1 Implications for predictability effects

Now, how does this relate to predictability effects? We already pointed out that predictability effects which operate on relatively static linguistic representations, such as the phonological string associated with a lexical entry, lead to a rough negative correlation between log word frequency and word length. Similarly, predictability effects which are manifested in token-by-token variation lead to lower probability tokens receiving a more redundant coding. For example, a word that has low probability given its preceding bigram might be pronounced twice as slowly, leading to twice as many 10-millisecond windows of a cochleogram information for the listener. Such an encoding means that this low-probability word is less likely to be completely masked by, for example, a random loud sound. Evidence of predictability effects on a token-by-token basis is itself direct evidence that talkers are picking a code that reduces the amount of redundant encoding.

Now, when you reduce the amount of redundant encoding, you either remove too much redundant encoding, which leads to errors, or you end up with a coding scheme

that is closer to the optimal code. That is, if errors do not increase, you end up decreasing the average code length without increasing the error rate: your attempted communication rate improves without increasing the error rate. Since the error rate in regular conversation between adult competent speakers is very near zero (at least for such things as word recognition and parsing), we can take it as given that predictability effects increase error rates negligibly, if at all. Therefore, any predictability effects that do exist reduce redundant encoding in a way that does not lead to errors, and so any evidence for predictability effects is itself direct evidence that talkers are reducing words in a way that is (closer to) optimal in the sense of the Noisy Channel Theorem.

The Smooth Signal Redundancy hypothesis (e.g. Aylett and Turk, 2004; Turk, 2010) and the Uniform Information Density (UID) hypothesis (e.g. Frank and Jaeger, 2008; Jaeger, 2010) differ from the presentation here insofar as they prioritize the notion of communicating a constant rate of something. However, they differ slightly on what, precisely, is supposed to be constant, and the differences correspond to subtle mischaracterization about what the Noisy Channel Theorm states.

The Smooth Signal Redundancy hypothesis appropriately emphasizes the role of signal redundancy in speech. However, Aylett and Turk (2004) and Turk (2010) both *equate* redundancy and recognition likelihood, and interpret information theory as favoring a code in which "signal redundancy (probability of recognition) is evenly distributed throughout each utterance" (Aylett and Turk, 2004, p. 33); that is, they take information theory to say that an efficient code is one which maintains a constant rate of signal redundancy, and that signal redundancy is the same as recognition probability.[2] Clearly, a robust code will achieve a relatively constant recognition probability over the course of an utterance, but this is because it will achieve a recognition probability that is almost always 100%. There is nothing special about uniform recognition probabilities *per se*. It is also not the case that recognition probability is the same as signal redundancy; more ambiguous messages will receive much greater signal redun-

[2] Aylett and Turk (2004) and Turk (2010) both give the following "illustration" of why a uniform or "smooth" rate of information transfer would be optimal. Consider a sequence AB, where each element has a probability of recognition of 0.5, and a sequence CD, where each element has a probability of recognition of 0.25 and 0.75, respectively. They point out that the probability of recognition of AB is $0.5 \cdot 0.5 = 0.25$, while the probability of recognition of CD is $0.75 \cdot 0.25 = 0.1875$. Since the sequence with uniform probabilities has a higher likelihood of recognition, Aylett and Turk take this as an illustration that a smooth signal is optimal. However, this illustration doesn't say much. There's no reason to pick probabilities that sum to 1, because the probabilities are for different events: the recognition of the first element, and the recognition of the second element. We could just as easily consider a sequence EF where E has probability of recognition 0.9 and F has probability of recognition of 0.6. The probability of recognizing FG under this skewed distribution is 0.54, clearly higher than the probability of recognizing either AB or CD.

dancy to achieve the same recognition probability near 100%.

The UID hypothesis of Frank and Jaeger (2008) and Jaeger (2010), on the other hand, prioritizes communicating at a constant rate of information near the channel capacity. Jaeger notes that a noisy channel has a maximum rate of information transfer for codes with an arbitrarily small error rate (our familiar $\max(H(X) - H(X|Y))$ quantity), and posits that an optimal talker will attempt to come as close to that rate as possible without going over. Jaeger points out that information rate can be understood as message characters per signal character, and takes signal characters to be words. The optimal strategy is then presented as one which says an extra word if the talker is in danger of exceeding the channel capacity. For example, consider the pair of sentences from Jaeger (2010):

1. My boss confirmed we were absolutely crazy.

2. My boss confirmed that we were absolutely crazy.

Clearly, each sentence encodes the same message, but the second sentence has one more word: "that." Jaeger says that if the message characters per word is higher than the channel capacity in the first example, an optimal talker will use the second example, thereby "distribut[ing] the same amount of information over one more word" (Jaeger, 2010, p. 27). This suggests that *merely* slowing down the rate of information transfer is sufficient: the point is to spread message characters across more signal characters. However, if we *only* slow down the rate of information transfer, the noisy received signal is just as ambiguous as it would be if we had omitted the extra word. Moreover, simply slowing down the rate of information transfer increases the number of signal characters used for each message character without affecting the ambiguity $H(X|Y)$. This means that the adopted code is longer *without* decreasing $H(X|Y)$, leading to a guaranteed loss in communication rate. The inclusion of "that" in less-probable contexts is optimal because *it makes the received message less ambiguous*: it decreases $H(X|Y)$. The verb "confirmed" can be followed by many kinds of syntactic structures (nominal objects, sentential complements, adverbs, prepositional adjuncts, &c.), but "that" can appear only as the determiner of a noun phrase or as a sentential complementizer.

This focus on smooth signals or uniform information density also involves an unnecessarily strong assumption that talkers are capable of approximating the optimal

code.[3] This assumption is required, because the UID is supposed to occur when talkers are speaking so efficiently that they repeatedly bump against a ceiling on possible information transfer. Remember, however, that our nearly-optimal code for the 22-character message set approached optimality by coding pairs of message characters using chunks of 9 signal characters. All of the proofs about optimal codes hold as chunk lengths go to infinity. As human sentence processing integrates many different levels of linguistic analysis extremely rapidly and in an online fashion, much linguistic information must be encoded in relatively small chunks. It is accordingly unlikely that talkers continually bump into the theoretical maximum communication rate for a given noisy channel.

Predictability effects, however, remain evidence of a tendency towards optimal codes in the sense of the Noisy Channel Theorem. We do not need to presume that they actually approximate the code that achieves the maximum rate $\max(H(X) - H(X|Y))$, where the maximization is over all possible codes, especially since the optimal code is probably not amenable to rapid online processing anyway. As detailed at the beginning of this subsection, predictability effects do provide evidence of tendency to minimize $H(X|Y)$, where possible, which in turn increases the actual information rate $H(X) - H(X|Y)$.

2.5 Conclusion

Section 2.4 closed by arguing that, under the hypothesis that predictability effects are about optimal communication, we would expect a negative correlation between *signifique* log probability and *signifier* length, but probably would not expect a signal that was actually "smooth" or attained "uniform information density." However, existing work, including that explicitly in support of the Smooth Signal hypothesis (e.g. Aylett and Turk, 2004; Turk, 2010) or the Uniform Information Density (UID) hypothesis (e.g. Frank and Jaeger, 2008; Jaeger, 2010) has in practice sought only to show this negative correlation. Their conclusions are, accordingly, still as valid as ever.

However, even with this negative correlation clearly demonstrated in a number of linguistic systems, it is still not obvious that predictability effects have much to do with efficiency. Predictability effects seem to move speech towards greater efficiency; it is not clear *how much* they move speech towards greater efficiency. Is the achieved

[3] Whether, as Jaeger (2010) points out, that optimal code is for arbitrarily few errors or an error rate arbitrarily close to some "acceptable" error rate

communication rate 50% higher due to predictability effects? Or is it only a tenth of a percent higher? In the former case, predictability effects make a clear contribution towards communicative efficiency, while the latter case suggests that predictability effects are just a curiosity of how brains and articulators make speech.

The most obvious way to determine which of these possibilities holds would be to simply measure how much predictability effects improve communication rates. Unfortunately, measuring the relative change in communication rates would involve finding very precise estimates for $H(X)$ and $H(X|Y)$ with and without predictability effects. This is clearly an extremely difficult problem. Chapter 4 focuses on presenting evidence that predictability effects exist in child-directed speech. One side effect of this exploration, however, will be establishing that predictability effects in CDS are not identical to those in adult-directed speech (ADS). This secondary result provides evidence that talkers perform some degree of listener modeling in picking codes that make $H(X|Y)$ small; a follow-up investigation into effects of eye-contact will suggest that talkers also pay attention to channel characteristics in picking codes that reduce $H(X|Y)$. Together, these secondary results will provide a new kind of evidence that predictability effects really are about communication, and not just some side effect with little appreciable impact on communicative efficiency.

Chapter 3

Bootstrapping Accounts

3.1 Introduction

Bootstrapping accounts, broadly speaking, propose that children use one kind of knowledge about language to help them gain a different kind of knowledge about language. For example, it has been proposed (see Section 3.2.1) that children use knowledge about a verb's meaning to help them learn whether it can be used in a passive construction. Similarly, the "Predictability Bootstrapping" hypothesis of this dissertation proposes that children use knowledge about linguistic reduction to help them learn other kinds of linguistic knowledge (and evaluates, using computational models, the specific proposal that children use knowledge about phonetic reduction to learn about syntax).

The term "bootstrapping" has been used in different ways by different authors in language development. This chapter aims, in part, to distill a useful technical definition for "bootstrapping account" out of the various usages. Specifically, this chapter advocates for a view of bootstrapping accounts as accounts that propose that two linguistic systems are correlated in a useful way; or, more precisely, exhibit systematic statistical dependencies that children use. This statistical dependency-centric view of bootstrapping accounts will motivate the use of explicit probabilistic models.

Section 3.2 clarifies why learning dependent systems together is easier than learning them independently, building the intuition that dependent systems live in a "smaller space" than a child would have to explore were she to assume they were independent. Section 3.2 proceeds to trace the historical development of bootstrapping accounts, reinterpreting them in this dimensionality-reduction view. Section 3.3 then describes and motivates the computational modeling philosophy of this dissertation as well-

suited for finding and measuring the statistical dependencies involved in bootstrapping. Towards this end, Section 3.3 closes with an overview of the Dependency Model with Valence, which is the basis of the models of Chapter 6, to help build intuitions about what statistical dependencies between syntactic analyses, syntactic grammars, and other discrete linguistic objects can look like.

3.2 Bootstrapping as dimensionality reduction

This section explains how a dependency between linguistic variables could reduce the work a child must do to learn the system producing those variables, and re-interprets previous bootstrapping accounts under this view.

Figure 3.1(a) presents a cartoon example of samples from two random variables that take on values between -1 and 1 and have a roughly linear correlation, a simple kind of statistical dependency; perhaps one variable is cartoon syntax, and the other is cartoon semantics. The cartoon language acquisition task, then, is to find a syntactic grammar that produces the right cartoon syntax points x, and a semantic grammar that produces the right cartoon semantics points y. If we were to pay attention to each variable in isolation, we would project each point to the vertical and horizontal axis. This projection would result in observed data with an apparently uniform, random distribution over the horizontal axis, and another apparently uniform, random distribution over the vertical axis. A successful learner that ignores the dependency between the produced variables would thus find systems that produce points uniformly at random along both axes, and would not reliably find systems that produce the right linear correlation between cartoon syntax and cartoon semantics. If our learner attends to the dependency between the produced variables, however, the search task becomes much more constrained; she wants to find systems that not only produce uniform distributions over both axes, but also produce a roughly linear correlation between the produced variables.

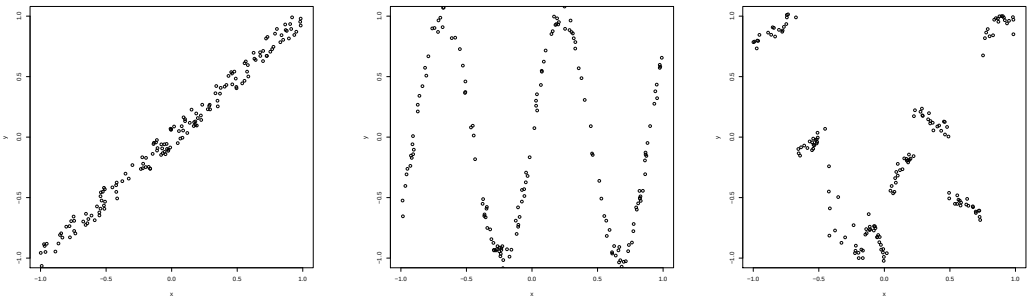
Let's be more precise about what we mean by "finding" a system that produces a particular set of points. If we pick some formalism for representing our syntactic knowledge, we can view each possible grammar as a point in some space with M dimensions.[1] For example, if we are using a Context Free Grammar, we would have

[1] Clearly, M may be infinite, but that is not a show-stopper here. Inference procedures exist for infinite-dimensional models, and, while the dimensionality-reduction intuitions are easier to develop when considering finite models, they carry over to the infinite-dimensionality cases.

one dimension for each possible CFG rule. For a particular grammar, each dimension would be 1 if the grammar contains that rule, and 0 otherwise. Alternatively, in a probabilistic approach, each dimension could be the probability of the corresponding rule. In both cases, each possible grammar would be a point in this M-dimensional space, and "finding" the right grammar involves exploring the different points in this M-dimensional space for the right one (or estimating a good probability distribution over such points). Similarly, we could have a semantic grammar space with N dimensions for N possible, e.g., λ-expressions. Combined grammars for both syntax and semantics are then situated in a "joint system space" with $(M+N)$ dimensions.

Now, because our observed cartoon syntax and cartoon semantics variables are dependent, we can mostly specify the location of a particular point by indicating its position along the line; rather than specify the cartoon syntax-and-semantics point $(-0.5,-0.5)$ with the semantics value and the syntax value, we can mostly specify it with a single number: $-0.5\sqrt{2}$ along the best-fit line. Because we can mostly specify the location of these points with a single number in this way, their intrinsic dimensionality is close to 1, not 2. Similarly, although the joint system space has $(M+N)$ dimensions, if there is a dependency between the M-dimensional syntax-generating space and the semantics-generating N-dimensional space, then the intrinsic dimensionality of the full syntax-semantics space is less than $(M+N)$. Bootstrapping accounts highlight cases where this dependency should be strong, thus significantly reducing the dimensionality of the joint system space. This reduction helps learning, because an infant learner can explore a low-dimensional space more quickly than a high-dimensional space: heuristic search procedures have less space to explore, and statistical approaches will obtain less sparse probability distributions over the reduced space.

Clearly, linguistic phenomena, and the grammars that underlie them, are not real-valued scalars between -1 and 1, but instead are complex objects with both discrete and continuous components. To understand how the dimensionality-reduction view applies to linguistic objects, consider that statistical dependencies can have different functional forms. The functional form of the dependency in Figure 3.1(a) is linear, but the functional form of the dependency in Figure 3.1(b) is sinusoidal. The intrinsic dimensionality of the data in Figure 3.1(b) is also close to 1, because the position of a point can be mostly captured by indicating its position along the sinusoid. Figure 3.1(c) presents data whose dependency has an even more complex linear and sinusoidal piecewise functional form, but can still be mostly specified by indicating a



(a) Linear correlation. (b) Sinusoidal correlation. (c) Arbitrary correlation.

Figure 3.1: Correlations with three different functional forms.

point's position along the function. Bootstrapping accounts, then, propose dependencies with linguistically-relevant functional forms. For example, some instantiations of Syntactic Bootstrapping hypotheses (discussed shortly) propose a functional form that relates the count of noun phrases to categorical semantic values.

Early presentations of bootstrapping accounts assumed a kind of directionality; system A is learned (or innately specified) first, and then features of system A are used to learn system B. For example, Semantic Bootstrapping (discussed shortly) assumes that the child obtains some kind of basic semantic representation of utterances or words first (from genetically-specified knowledge and prelinguistic event and concept representations), and then uses features of that basic semantic representation to help learn features of the syntactic representation. While there's no reason to assume this kind of directionality in general (and we'll see shortly that more recent presentations avoid it), particular instances may yet have an inherent directionality. For example, consider again the sinusoidal dependency of Figure 3.1(b): each x value is associated with only one fairly tight region of y values, but each y value is associated with 2-5 regions of x values. Thus, knowing x is useful for making inferences about y, but y is less useful for making predictions about x.

The functional form of the dependency is also relevant to questions of nativism and empiricism. Returning to the cartoon syntax and semantics example, if our infant is prepared to consider only linear correlations, the sinusoidal dependency of Figure 3.1(b) will be less useful because the infant will waste resources exploring grammars that produce points between peaks and troughs. Accordingly, if the relevant dependency is highly complex, we may be driven to a more nativist position; a purely empiricist approach that always considers every possible functional form for statistical

dependencies is likely too unconstrained to succeed (although inference procedures do exist that are capable of discovering highly non-linear dependencies, such as kernelized principle components analysis and spectral clustering).

We will shortly discuss specific bootstrapping accounts, and see that they have primarily been evaluated using laboratory experiments. Computational models, however, can provide important complementary evidence. Bootstrapping accounts rely on the existence of a useful dependency; while laboratory experiments can show whether children attend to the relevant correlates in the right way, computational models can measure the strength and utility of the putative dependency in the evidence children have under explicit assumptions about the functional form of the dependency. Thus, even if children use some set of heuristics rather than a probabilistically-optimal inference procedure, a probabilistically-optimal system can reveal whether a particular proposed dependency is strong enough to bring down the dimensionality of the space the grammar lives in, facilitating any kind of learning. That is, our computational models will be about the shape of the data, not a proposal for what children procedurally do.

Finally, before turning to individual bootstrapping accounts, it should be noted that, under this view, a bootstrapping account differs from other kinds of learning accounts only in that a bootstrapping account proposes the learner uses features that linguists regard as coming from different linguistic systems. Learning within a system relies on dependencies to reduce the dimensionality of the system in the same way; for example, syntax acquisition is possible because, e.g., the distributional regularities of one verb are informative about the distributional regularities of other verbs, since verbs appear in broadly similar environments, and also about the distributional regularities of, e.g., nouns, since verbs appear in broadly different environments from nouns.

Next, we proceed to discuss specific, well-known bootstrapping accounts, re-interpret them in the dimensionality-reduction view, and then present Predictability Bootstrapping as a kind of bootstrapping that should be especially easy.

3.2.1 Semantic Bootstrapping

One of the earliest bootstrapping accounts proposed that infants learn syntax by first getting a semantic representation of utterances, and then mapping that representation into a rudimentary syntactic analysis. For example, Bruner (1975) posited that "the infant first learns pre-linguistically to make the conceptual distinctions" that are linguis-

tically relevant, and begins the process of syntax acquisition by mapping conceptual structures over things the infant perceives to be happening into syntactic structures over words the infant hears. Pinker (1984) proposed, similarly, that infants posit syntactic structures as a reflex of the observed semantics, with some syntax-semantics correspondences innately specified, at some level of abstraction, in Universal Grammar.

As a concrete example, consider Pinker et al.'s (1987) proposal for the acquisition of the passive alternation, a type of syntactic regularity. Some verbs, such as "weighed," can be passivized: "Dr. Caron weighed the patient" and "the patient was weighed by Dr. Caron" are both fine. Other verbs, however, cannot: "Pat has three bicycles" is fine, but "three bicycles are had by Pat" is at least very weird. How do children learn which verbs can and cannot be passivized? Using longitudinal corpus data, Pinker et al. (1987) concluded that children do passivize verbs that cannot grammatically be passivized, so they are not simply conservative in their production.

In particular, Pinker et al. (1987) proposed that children exploit a structured dependency between thematic roles and passivizability, involving a tripartite division of verbs:

1. Canonically actional verbs, which denote actions and have agent subjects and patient objects.

2. Anti-canonically actional verbs, which denote actions and have patient subjects and agent objects.

3. Non-actional verbs, which do not denote actions and may have a variety of thematic role assignments.

Under this division, Pinker et al. posited that canonically actional verbs are always passivizable, anti-canonically actional verbs are never passivizable, and non-actional verbs are sometimes passivizable. Pinker et al. argued that children learn the passivizability of non-actional verbs by first learning different verb subclasses on a language-by-language basis. Each subclass has similar basic meanings and the same thematic role-assignments (such as the verb classes of Levin (1993)). For example, we might propose an English subclass of causation of motion (such as "kick"), and an English subclass of change of possession (such as "give"). Also on a language-by-language basis, we allow a verb subclass to be associated with and "inherit" the thematic role assignment of another subclass. For example, English allows verbs of causation of

motion to inherit change-of-possession thematic role assignments, licensing sentences such as "he kicked/tossed/slid her the ball."

Pinker et al. called these associations "conflation classes," and proposed that those non-actional verbs which can be passivized are exactly those which are in a conflation class with actional verbs. Since actional verbs are always passivizable, such non-actional verbs are accordingly capable of inheriting passivizability. The acquisition of the passive, under this account, involves a (possibly innate) principle that actional verbs are always passivizable, learning of verb subclasses in terms of both basic verb meaning and thematic role assignment, and learning inheritability associations between verb subclasses.

Under the dimensionality-reduction view, one of the variables in this account is (something like) the syntactic environments of observed verb instances, some of which will occur in passive structures and some of which will not. The other variable is the semantic class and thematic role assignment of the verb instances. Whether the verb class occurs in passive structures is then a dimension of the (syntactic) grammar, which should have some notion of the passive alternation, and the semantic class and thematic role assignment of the verb are dimensions of the semantic system, perhaps simply a lexicon. Finally, the conflation class constraint on passivizability simply defines a highly complex functional form for the dependency between the semantics of verbs and the syntax of verbs. Given the complexity of this functional form, it probably must be innately specified to be useful, as advocated by Pinker et al. (1987).

3.2.2 Syntactic Bootstrapping

The Semantic Bootstrapping hypothesis just discussed proposes that semantic knowledge may help with the acquisition of syntactic knowledge, and the Syntactic Bootstrapping hypothesis we now turn to proposes syntactic knowledge may help with the acquisition of semantic knowledge. Specifically, Gleitman (1990) proposed that infants exploit the fact that words with similar syntax often have similar meanings. Beyond the directionality of inference, Syntactic Bootstrapping accounts differ from Semantic Bootstrapping accounts in the complexity of the dependency they rely on. Specifically, Semantic Bootstrapping accounts tend to propose highly complex dependencies that involve a lot of complex, innate knowledge (as we saw in the previous subsection), while Syntactic Bootstrapping accounts tend to focus on simpler dependencies that involve more easily-learned knowledge.

For example, Gleitman points out that verbs whose meaning involves transferring an object from one place to another will generally take as arguments one noun phrase for the entity doing the transfer, one noun phrase for the object transferred, and one noun phrase for the destination. Moreover, the entity causing the transfer will typically be in subject position, while the other noun phrases will be direct and indirect objects, respectively: "Pat paid five dollars to Robin," "Jesse put the ball in the basket." Thus, one of the variables is whether a sentence's meaning involves transfer, the other variable is the number of noun phrases in the sentence, and the proposed statistical dependency associates sentences with three noun phrases with meanings of transfer. Several laboratory experiments have since supported the idea that children make semantic inferences on the basis of syntactic cues, such as the inference that a novel verb is causitive if it is used transitively (Naigles, 1990; Fisher, 1996), that a novel verb is telic (i.e. with a set goal, result or outcome) if it is used transitively (Wagner, 2010), and that a novel word is a preposition if it occurs before a noun phrase and after a verb (Fisher et al., 2006).

Although Gleitman proposed Syntactic Bootstrapping as an alternative to Semantic Bootstrapping, in fact the two are mutually compatible. Infants may use whatever syntactic cues they get to learn about semantics, and whatever semantic cues they can get to learn about syntax. From the perspective of probabilistic modeling, this just corresponds to learning a joint model over syntactic and semantic representations, rather than a conditional model of syntax on semantics (Semantic Bootstrapping) or of semantics on syntax (Syntactic Bootstrapping). Kwiatkowski et al. (2012) provide just such a computational model. Their model learns a Combinatory Categorial Grammar (CCG) lexicon from utterances (transcribed word sequences) paired with their meanings by modeling probability distributions over hidden CCG derivations. Evaluated on child-directed speech, they show that their model correctly learns both syntactic and lexical semantic information about English, including that English is SVO and the meanings of some quantifiers.

3.2.3 Prosodic Bootstrapping

The focus of this dissertation is exploring the role of predictability effects in language use and acquisition, but predictability effects are not the only source of suprasegmental variation. This section discusses prosody (roughly, the structure of speech as conveyed by rhythm and intonation) and how it might be useful for syntax acquisition by chil-

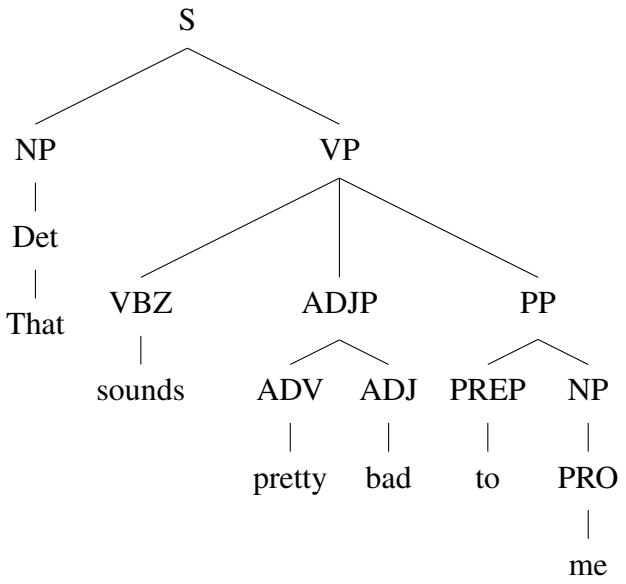


Figure 3.2: Example constituency tree.

dren. Specifically, it has been proposed for quite some time in the "Prosodic Bootstrapping" hypothesis (e.g. Gleitman and Wanner, 1982) that prosody would be useful for syntax acquisition. Other than the work described in this dissertation, however, we are not aware of a computational evaluation of this hypothesis. We will now take a basic overview of syntactic representations and prosodic theory in preparation for a precise presentation of Prosodic Bootstrapping.

3.2.3.1 Syntactic Representations

We will ultimately be learning probability distributions over syntactic representations, so it is important to consider the statistical ramifications of different kinds of syntactic representations. This dissertation considers two kinds of syntactic representations.

One representation we consider is constituency structure. Constituency structure models syntactic structure as a hierarchical tree of different phrases. Figure 3.2 presents an example constituency tree, and we can see that it has a node for a Verb Phrase spanning the words "sounds pretty bad to me," and inside this Verb Phrase are further phrasal nodes for an Adjective Phrase spanning "pretty bad" and a Prepositional Phrase spanning "to me."

The defining feature of a syntactic constituent is substitutability: subtrees whose top nodes share the same label can be substituted for each other. The final NP, for example, could be replaced by the much longer NP in Figure 3.3, producing the perfectly

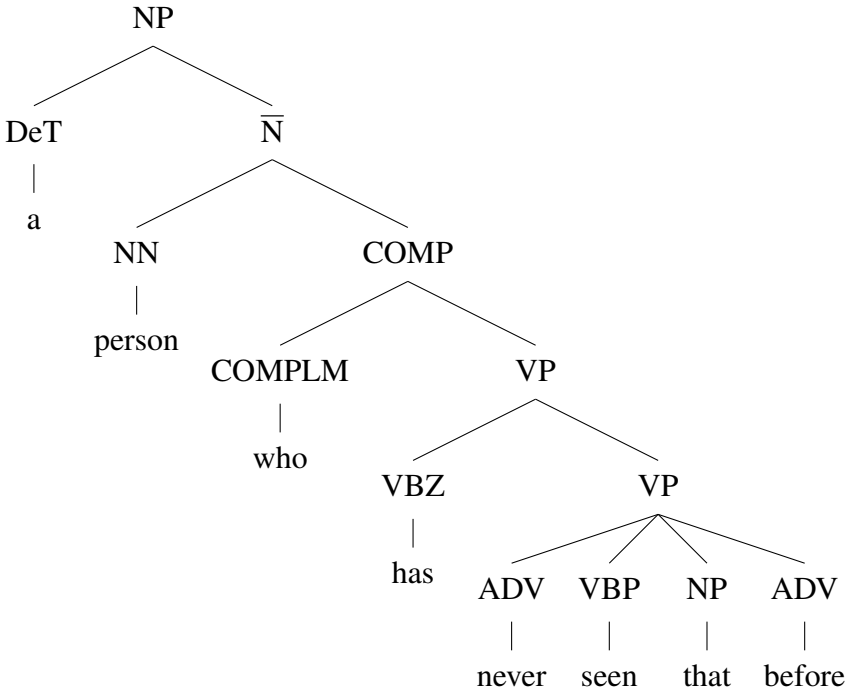


Figure 3.3: Large noun phrase.

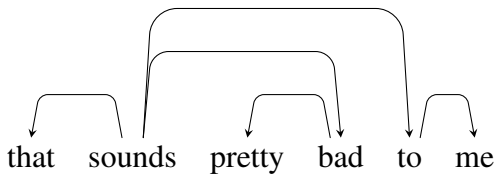


Figure 3.4: Example dependency tree.

natural "That sounds pretty bad to a person who has never seen that before."

Constituency-based syntax focuses on dominance and precedence relations, mostly between unobserved phrasal nodes rather than observed lexical nodes. So, an NP can be a Determiner, it can be a Pronoun, it can be a Determiner followed by an $\overline{\text{N}}$, but it cannot be a $\overline{\text{N}}$ followed by a Determiner.

The other representation we consider is dependency structure, which focuses on describing which words are modified by which words. It does so by, like constituency grammar, positing a tree structure over the observed words. However, dependency structures have no phrasal nodes: every arc is between two words.

Figure 3.4 presents an example dependency tree for the same sentence as in Figure 3.2. As we can see, each arc has an arrow at one end. The word at the arrow

end is called the dependent of the arc, and the word at the tail of the arc is called the head (Mel'čuk, 1988). So, for example, "that" is the dependent of "sounds" because "that" is the subject of "sounds," and "to" is a dependent of "sounds" because "to," a preposition, is the head of a prepositional phrase modifying "sounds."

3.2.3.2 Prosodic Theory

Prosody is a theoretical linguistic concept positing an abstract organizational structure for speech.[2] While it is often closely associated with such measurable phenomena as movement in fundamental frequency or variation in spectral tilt, these are merely observable acoustic correlates that provide evidence of varying quality about the hidden prosodic structure, which specifies such hidden variables as contrastive stress or question intonation.

Although there are several theories of how to represent and annotate prosodic structure (e.g. Nespor and Vogel, 1986; Selkirk, 1984; Liberman and Prince, 1977), one of the most influential is the theory underlying the ToBI (Tones and Break Indices, e.g. Beckman and Pierrehumbert, 1986; Beckman et al., 2005) annotation scheme. This discussion will use examples from ToBI both because ToBI brings out important issues in prosodic theory and because ToBI annotations will be used as input for experiments in Chapters 5 and 6. ToBI proposes that the prosodic phrasing of languages is structured into nested and ranked phrase types, and that phrasal phenomena are cued by durational and intonational regularities. For example, Mainstream American English (MAE) ToBI proposes that there are two types of prosodic phrase for this language variety. An *Intonational Phrase* is the "biggest" kind of phrase, and can contain one or more *Intermediate Phrases*. The end of an Intonational Phrase is cued by severe lengthening of the last word's rime (last syllable coda), along with the presence of certain tones (catogorical pitch events). The end of an Intermediate Phrase, on the other hand, is cued by less severe lengthening of the last word's rime, and a different set of tones. Finally, each intermediate phrase can contain one word that bears a pitch accent.

This kind of prosodic phrase structure implements a pair of principles which are together called "strict layering." First, strict layering posits that there are a small number of prosodic phrase categories (in MAE ToBI, there are two: Intonational and Intermediate). Second, strict layering posits that the prosodic phrase categories are ranked, such that a prosodic phrase of some type can contain prosodic phrases of a

[2] Signed languages also exhibit prosodic phenomena, but they are not addressed here.

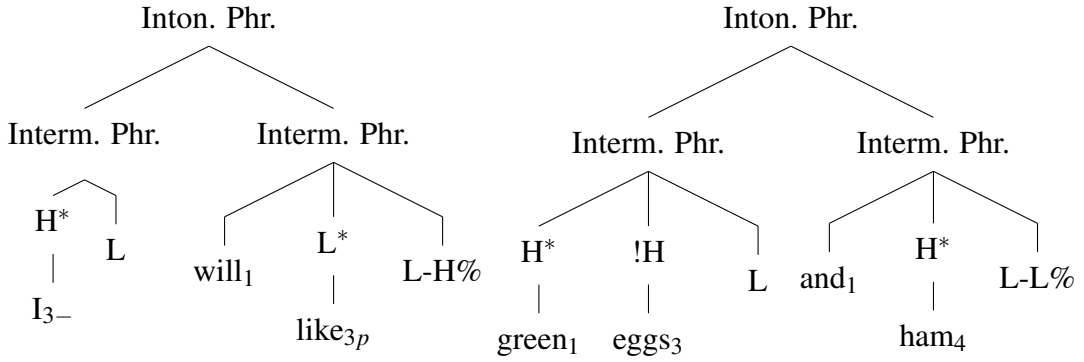


Figure 3.5: Possible ToBI analysis for "I will like green eggs and ham," with break indices subscripted.

lower-ranked type (in MAE ToBI, Intonational phrases are ranked above Intermediate phrases). This principle enjoys broad acceptance beyond ToBI (e.g. Nespor and Vogel, 1986; Selkirk, 1984; Liberman and Prince, 1977).

Figure 3.5 presents a ToBI analysis for one possible pronunciation of "I will like green eggs and ham." Reasonable tone categories (the L and H annotations) have been included for completeness and to reinforce the fact that ToBI uses both word duration and intonational facts in constructing prosodic phrase structures. However, the focus of this work is word duration, and intonation will not be discussed further.

Note that each word is subscripted with a number. These numbers are called "break indices," and indicate both the strength of the break between the words and the position of the word in prosodic phrase structure. One of the claims of the theory behind ToBI s that the prosodic phrase structure can be recovered from the sequence of break indices, and that they can be recovered from the intonational and durational phenomena. A break index of 4, for example, corresponds to the end of an Intonational Phrase, an index of 3 corresponds to the end of an Intermediate Phrase, an index of 1 corresponds to a normal word index, and an index of 0 corresponds to the boundary between a clitic and its base word. Break indices of 2 are used to indicate phrasal weirdness (an Intermediate Phrase break that seems to be missing its boundary tone, for example), and break indices can be annotated with "-" to indicate "weaker than usual" or with "p" to indicate that the break is accompanied by a pause.

3.2.3.3 Prosodic Cues to Syntax

Prosody has been hypothesized to be useful for learning syntax because the kind of prosodic constituency structure we saw above sometimes aligns with traditional con-

stituency analyses (Ladd, 1996; Shattuck-Hufnagel and Turk, 1996). For example, Selkirk (1978) points out that in the sentence:

- In Pakistan, Tuesday, which is a weekday, is, Jane said, a holiday.

most phrases delimited by commas are syntactic constituents, and all of them are prosodic constituents.

There are four basic results from the literature on this subject that suggest prosodic structure could be useful for infants learning language. First, adults will use prosodic information to disambiguate syntax in a laboratory environment, such as a word's Part-Of-Speech (e.g. in French Millotte et al., 2007) or phrasal bracketing (such as high compared to low PP-attachment and identification of parentheticals, e.g. Price et al., 1991), so children must learn the correspondence at some point.

Second, it has been shown that infants are sensitive to prosodic manipulations in performing syntactic tasks. For example, Seidl (2007) investigated, using a headturn procedure, how infants take advantage of suprasegmental cues, such as pauses, word duration, and fundamental frequency, to identify sentence-internal clause boundaries. Specifically, they first recorded a naïve talker saying the same sequence of words either as a clause or not a clause. For example, for the stimulus "rabbits eat leafy vegetables," the clause version was taken from a recording of "Many animals prefer green things. Rabbits eat leafy vegetables." The non-clause version was taken from a recording of "John doesn't know what rabbits eat. Leafy vegetables taste so good." Six-month-olds were familiarized to both the clause and non-clause versions in a familiarization stage, and then played the full passages in a test phase. In a replication of Nazzi et al. (2000) and Soderstrom et al. (2003), Seidl reported that infants listened longer to the familiarized test passages in which "rabbits eat leafy vegetables" does not contain a clause boundary. Moreover, Seidl manipulated the acoustic signal, and found that infants continue to distinguish clausal stimuli from non-clausal stimuli even when post-clausal pauses were removed, but not when fundamental frequency or durational cues were removed. As prosodic phrasing is usually defined with respect to intonation and word duration but not pauses, these results together suggest that the infants were indeed relying on cues to a hidden prosodic structure.

Third, infants begin learning basic prosodic facts, such as the rhythmic properties of their language, very early. Specifically, Mehler et al. (1988) recorded a native bilingual French-Russian speaker talking in French and Russian, and then played the recordings to 4-day-old French babies while measuring how quickly they sucked on a

pacifier. They reported that the infants sucked faster for longer when listening to the French recordings. This effect persisted when the recordings were low-pass filtered, but disappeared when the recordings were played backwards. Mehler et al. pointed out that low-pass filtering eliminates most segmental information (e.g., the relative prevalence of voiced and voiceless sibilants in a language) but preserves most rhythmic information (e.g., the relative duration of consecutive vowels), while playing a recording backwards corrupts rhythmic information while preserving most segmental information. They thus concluded that infants must be attending to rhythmic cues in discriminating the stimuli. They also examined monolingual American English-learning two-month-olds in a similar set-up (using looking times rather than sucking rate) on American English and Italian stimuli, and reported essentially the same pattern of results. All together, these studies indicate that basic prosodic information is one of the very first aspects of language to be learned.

Fourth, Morgan et al. (1987) showed that adults will use (artificial) prosodic information in learning the syntax of an artificial language. Specifically, Morgan et al. first defined an artificial language with hierarchical structure. They then produced three kinds of spoken stimuli: monotone, inconsistent, and consistent. To produce the first, monotone kind, their talker spoke each sentence as if each word were just an item in a list. To produce the second, inconsistent kind, they bracketed groups of words that did not correspond to syntactic constituents, and the speaker read the sentences "in such a fashion that vowel lengthening, pitch discontinuities, and pausing served to group the words into units." To produce the third, consistent kind, they bracketed groups of words that did correspond to syntactic constituents, and their talker used duration, intonation, and pausing to indicate these consistent groups. They then assigned subjects to one kind of input, and exposed them to the spoken version of each sentence, an orthographic transcription, and a sequence of pictograms associated with each word. After exposure, subjects were tested on the vocabulary of the language, rules of the language, and various constituent/non-constituent discrimination tests. While subjects exposed to all three kinds of spoken stimuli succeeded in learning the basic vocabulary, Morgan et al. (1987) found that those exposed to the prosody consistent with syntactic structure performed much better when identifying rules and licensed constituents of the artificial language. All together, this indicates that prosodic phrasing information can be useful for learning about syntax, at least for learning the syntax of a small language.

There is also some computational work on using prosodic cues for parsing (e.g. Gregory et al., 2004; Kahn et al., 2005; Dreyer and Shafran, 2007; Nöth et al., 2000), but this work is all supervised, observing gold-standard syntax, gold-standard prosodic annotations, or both, so it does not bear directly on the question of whether prosodic structure is useful for *learning*. Nevertheless, it is encouraging that prosodic cues are useful for parsing in a supervised context; if they were not, then Prosodic Bootstrapping would probably be hopeless.

So there is solid laboratory evidence that infants both pay attention to acoustic information that cues prosodic structure and take advantage of that information, at least in laboratory syntactic tasks. Additionally, there is some computational evidence of reliable correspondences between prosodic structure and syntax, although it is not yet obvious if these correspondences are readily learnable in an unsupervised fashion. In any case, if prosodic structure is sufficiently prominent in the acoustic signal, and aligns often enough with syntactic structure, then it may provide children with important information about how to combine words into phrases early in the language acquisition process.

Under the dimensionality-reduction view of this bootstrapping account, one of the variables is syntactic trees, the other variable is prosodic phrase trees, and the proposed dependency is one of occasional identity between syntactic constituents and prosodic constituents, or between syntactic boundaries and prosodic boundaries. This formulation makes explicit certain hitherto-unappreciated computational difficulties with prosodic bootstrapping: because prosodic trees are flat, but syntactic trees are deep, the proposed dependency must be highly complex, illustratively closer to the arbitrary dependency of Figure 3.1(c) than to the linear correlation of Figure 3.1(a). This kind of difficulty could be circumvented by specifying the functional form of the dependency innately, as in Semantic Bootstrapping. Innateness is plausible for the case of Semantic Bootstrapping because speakers of different languages talk about broadly the same kinds of things under the same kinds of communicative constraints. Different languages can have very different prosodic systems, syntactic regularities, however, so it is unlikely that the functional form of the dependency underlying Prosodic Bootstrapping is innate.

3.2.4 Predictability Bootstrapping

This subsection describes Predictability Bootstrapping, a novel hypothesis introduced by this dissertation. It has already been established that language exhibits many kinds of predictability effects: linguistic forms tend to be longer and more distinct when what they represent is less probable. Moreover, such predictability effects exist both for relatively static components of grammatical knowledge, such as the inverse correlation between word frequency and citation phonological form, and for fairly dynamic components of grammatical knowledge, such as the duration of a particular word pronounced in a particular syntactic context. The Predictability Bootstrapping hypothesis proposes that predictability effects allow children to use the observed degree of reduction as a cue to the probability of a word and associated hidden linguistic structure.

To illustrate how Predictability Bootstrapping could help beyond syntax acquisition, let's briefly consider how it could help in another domain. Specifically, let's examine how it could help children with learning new words. Swingley and Aslin (2007) point out that children must be willing to learn words that are phonological neighbors, such as "sip" and "ship," but must also be able to tolerate phonetic invariance. If a child knows the word "ship," and hears it with the initial consonant much less palatalized than usual, should the child treat the heard word as a variant of "ship" or hypothesize a new word type? In a preferential looking task, Swingley and Aslin (2007) show that children do indeed exhibit difficulty learning novel words that are neighbors of known words.

However, children must ultimately succeed at this kind of task even if it is more difficult, and they could take advantage of predictability effects. If they hear a novel pronunciation of "sip" that is very slow, but they have independent evidence that "ship" is a common word (i.e. they've heard it relatively often), they could reason that the slow pronunciation of "sip" is evidence that it represents a rare word type, and so is probably distinct from the common word "ship." Now, there are many reasons a particular token might be pronounced slowly, so a child would not necessarily be able to conclude that "sip" is a new word after one instance, but it could help the child determine that it is a new word after hearing fewer instances or in conjunction with weaker corroborating evidence. Unfortunately, Swingley and Aslin use the same audio token for the novel words throughout their experiment, so this possibility could not be examined without gathering more data.

As previously mentioned, however, the computational models presented in this dis-

sertation focus on the bootstrapping of syntactic structure from predictability effects. Let's consider how, specifically, this could occur. Gahl et al. (2006) examined the pronunciation of verb phrases of different probabilities by naïve talkers. Specifically, they examined verbs that can be used either transitively or intransitively, but strongly prefer one use. For example, "dance" prefers to be used intransitively, but can be used transitively:

- Intransitive: When the radiant ballerina danced, the role became world-famous.
- Transitive: When the radiant ballerina danced the role, it became world-famous.

Similarly, "lost" prefers to be used transitively, but can be used intransitively:

- Intransitive: Even though the team lost, the match meant a big success for the new coach.
- Transitive: Even though the team lost the match, it meant a big success for the new coach.

Because "danced" has an intransitive bias, an intransitive verb phrase with "danced" should, *ceteris paribus*, have a higher probability than a transitive verb phrase with "dance." Similarly, a transitive verb phrase with "lost" should have a higher probability than an intransitive verb phrase with "lost." If talkers exhibit predictability effects with respect to syntactic probability, then talkers should reduce words in the high-probability verb phrases more heavily.

Gahl et al. (2006) found that talkers did, in fact, pronounce words in the high-probability verb phrases more quickly. Specifically, they found that the verbs themselves were pronounced more quickly in the high-probability (i.e. bias-matching) verb phrases. Moreover, they reported a trend wherein the direct object noun phrase of the transitive uses was pronounced more quickly when the verb was biased towards transitive uses (although this comparison was significant only by subjects, not by items, in a traditional ANOVA). So transitive and intransitive frame probabilities exhibit predictability effects. Additionally, this study is an extension of Gahl and Garnsey (2004), which found similar patterns for verbs that are biased to take either a Direct Object or a Sentential Complement, suggesting that predictability effects may exist for a variety of syntactic phenomena. Tily et al. (2009) found that these effects are not restricted to a laboratory environment, reporting that a corpus of spontaneous speech exhibited the same tendency.

Now let's consider how this might be useful for a child trying to learn to talk. Suppose a child has independent evidence that "dance" prefers to be used intransitively (perhaps she has heard it often at the end of a sentence), and then hears "dance" pronounced slowly in the middle of a sentence. She could take the slow pronunciation as evidence that "dance" is probably being used transitively. From a constituency parsing perspective, she could then allocate less probability mass to parses that insert several constituent boundaries between "dance" and possible rightward arguments. Alternatively, from a dependency parsing perspective, she could allocate more probability mass to parses which give "dance" a rightward argument.

3.2.4.1 Predictability and Prosodic Bootstrapping compared

How does Prosodic Bootstrapping compare to Predictability Bootstrapping? It is important to point out that they are mutually compatible. It is entirely possible that infants pay attention to both prosodic structure and signal redundancy in language acquisition. However, they are different mechanisms and so do make different predictions, which will be examined in the computational modeling experiments of Chapters 5 and 6. First, they require the infant to deal with different kinds of ambiguity. To succeed at prosodic bootstrapping, the infant must successfully parse prosodic constituents using acoustic cues, which is not a trivial task, and then learn which prosodic constituents do and do not correspond to syntactic constituents. Steedman (1996) argues that prosodic constituents that do not line up with traditional syntactic constituents may nevertheless correspond to the non-traditional constituents that arise in some Combinatory Categorial Grammar derivations. If this kind of correspondence holds robustly, it would simplify the functional form of the statistical dependency substantially, as the infant would only need to allow for the possibility that incorrectly parsed prosodic constituents do not line up with syntactic constituents. In any case, Prosodic Bootstrapping is hard because it involves successful induction of and inference over an intermediate hidden prosodic phrase structure representation.

Predictability Bootstrapping, on the other hand, does not involve any intermediate representation: words in a highly predictable syntactic environment are, *ceteris paribus*, pronounced more quickly. However, as many factors beyond syntax are considered when determining the predictability of a word, and moreover many factors beyond predictability are considered when determining the duration of a word, there is considerable ambiguity over the cause of a word's duration; perhaps it is a First-Mention (and so *ceteris paribus* longer) but also highly probable in its syntactic context

(and so *ceteris paribus* shorter).

Finally, Prosodic Bootstrapping is only really natural when considered as a strategy for learning *constituency* structure, while Predictability Bootstrapping applies in any situation involving probabilities. This is important because, as Chapter 6 details, the more successful approaches to unsupervised grammar induction have typically used dependency rather than constituency representations. Chapter 5 compares Prosodic Bootstrapping with Predictability Bootstrapping in a constituency-based grammar induction model, and Chapter 6 compares the two approaches in a dependency-based grammar induction model.

3.2.5 Formulating and evaluating bootstrapping accounts

The introduction to this chapter hinted that the original presentation of these bootstrapping accounts did not reify statistical dependencies as the defining feature of a bootstrapping account in quite this way. Indeed, Pinker (1984) used the term "bootstrapping" not in reference to a broad class of language development accounts but in reference to a specific problem in constituency syntax acquisition: namely, the acquisition of unobserved phrasal categories, which was alleged to be logically circular[3] because each such category is defined distributionally in terms of other unobserved phrasal categories. In Pinker's original account, semantic information was used to break the circularity of this "bootstrapping" problem, but the appeal to semantics for the acquisition of syntax was not the original motivation for the term "bootstrapping."

Defining "bootstrapping accounts" in terms of a specific statistical dependency allows us to be more precise in discussing the accounts. Morgan and Demuth (1996), for example, criticized the use of the phrase "prosodic bootstrapping" to refer to accounts that relied on some relationship between *suprasegmental* cues and syntax, not between prosodic structure *per se* and syntax. They suggested the term "phonological bootstrapping" for such accounts instead. Their suggestion anticipates this chapter's reification of statistical dependencies as the defining feature of bootstrapping accounts. Specifically, a "phonological bootstrapping" account is one that relies on any kind of statistical dependency between suprasegmental cues and the target knowledge (typically syntactic knowledge), and a "prosodic bootstrapping" account is one in which

[3]This kind of circularity persists in statistical approaches, and is a general problem for any model that includes statistical dependencies between unobserved variables. However, as Section 3.3.2.1 will mention briefly, a statistical formulation allows approximate numerical solutions, and so is not fatally circular in a logical sense.

the functional form of this statistical dependency refers to prosodic structure. Strictly speaking, the "prosodic bootstrapping" account presented above is even more specific, as it further constrains the functional form to refer only to occasional alignment between prosodic phrases (or boundaries) and syntactic phrases. We could thus call this specific account the "prosodic alignment bootstrapping" account.

This view clarifies why we have chosen to use the name "predictability bootstrapping." At first glance, "predictability bootstrapping" seems like an outlier among bootstrapping account names: prosodic, syntactic, and semantic structures are all lingustic variables, while "predictability effects" refers to a specific relationship between variables. If we name bootstrapping accounts according to the statistical dependency they rely on, however, it is perfectly natural to refer to a specific relationship between variables. Thus, predictability bootstrapping accounts form a subset of what we might call "redundancy bootstrapping" accounts, which would propose that *some* relationship involving signal redundancy is important for language acquisition, but would not make claims about what that relationship looks like.

This new nomenclature will make the computational modeling strategy pursued in Chapters 5 and 6, and introduced shortly, more transparent. Specifically, the goal of our probabilistic models will be to characterize what kinds of statistical dependencies exist between word duration and local syntax. To this end, those chapters will design models that implement "durational bootstrapping" of syntax; they will look for a wide range of statistical dependencies between word duration and local syntax in the data. We will then examine the *specific* statistical dependencies that are evident in the data. To the extent that these statistical dependencies are mediated by prosodic structure, the models will indicate that prosodic bootstrapping is plausible in the sense that prosodically-mediated evidence about syntax exists in word duration patterns; to the extent that these statistical dependencies reflect a negative correlation between word duration and syntactic probability, the models will indicate that predictability bootstrapping is plausible.

3.3 Computational Models

As discussed in Section 3.2, bootstrapping accounts come down to a proposal that a particular dependency between two different kinds of linguistic knowledge reduces the dimensionality of the space in which the target grammar "lives," thereby simplifying the language acquisition process for children. While bootstrapping accounts

have primarily been evaluated using laboratory methodologies, the dimensionality-reduction view shows how computational models can provide important complementary evidence. Specifically, laboratory studies can show whether children respond to the covariates in question in the right way, and computational methods can evaluate the strength and practical utility of the putative dependency in real language input. Chapters 5 and 6 develop computational models that are suitable for discovering and exploiting dependencies between word duration and syntax. These models will allow us to compare the Predictability Bootstrapping hypothesis and Prosodic Bootstrapping hypothesis for syntax.

This section introduces the basic modeling philosophy of this dissertation, motivates the use of explicit probabilistic models in comparison to alternative approaches for the purposes of this dissertation, and provides an intuitive introduction to the model of syntax that will form the core of the models in Chapter 6.

3.3.1 Modeling philosophy

The computational modeling philosophy of this dissertation is best understood in the context of Marr's levels of analysis (Marr, 1982). Marr identifies three kinds of models for an information-processing system, using a cash register as an example information-processing system: the "computational" level, the "algorithmic/representational" level, and the "implementational" level. Computational-level analyses seek to characterize the overall problem the information-processing system seeks to solve; for the cash-register, the problem is the total charge of the purchase, and an appropriate computational-level theory is the theory of addition. There are many ways to represent and compute sums, and an algorithmic/representational-level theory may specify that the cash register uses a binary, rather than decimal, representation for numbers, and keeps a running total rather than storing each item's price and summing the charge at the end. Finally, different physical implementations can realize the same algorithm manipulating the same representations. An implementational-level theory of a cash-register would specify, e.g., how different patterns of transistors perform the addition.

This dissertation seeks to address computational-level questions regarding syntax acquisition. Computational-level approaches are attractive because they seek only to characterize the problem that children must solve, without making strong commitments about *how* the child solves the problem. Other modeling approaches that have been adopted to address the development of syntax (Bannard et al., 2009; Chang et al.,

2006b; Freudenthal et al., 2006, 2007) propose that children attend to statistical properties of the input, but also make commitments about how those statistical cues are represented and tracked. For example, the syntactic component of the Dual-path model of Chang et al. (2006a) is a Simple Recurrent Network that is trained to predict the next word on the basis of previous words. While this model proposes that children attend to statistical regularities in the input, it captures only those statistical regularities in the preceding history that are useful for predicting the next single word. Our models, by contrast, will gather statistics that are relevant for disambiguating the assumed form of the unobserved syntax.

Specifically, the models in Chapter 5 will characterize a probability distribution over chunking tag sequences $\mathbf{c}$ that correspond to a shallow constituency parse, given a corpus of spoken utterances C, $P(\mathbf{c}|C)$. Similarly, the models in Chapter 6 will characterize a probability distribution over dependency trees $\mathbf{t}$ given a corpus of spoken utterances C, $P(\mathbf{t}|C)$. To evaluate these models, we will use them to identify the highest-probability parses for new utterances, and compare the parses under the models to hand-annotated parses. Additionally, we will compare the models to simple baselines that capture superficial aspects of the data.

This style of evaluation, which is standard in the field of Natural Language Processing (NLP), is at odds with the kind of evaluation paradigm often adopted in the field of cognitive science for computational models of syntax. The Dual-path and MOSAIC models, for example, were both evaluated by seeing how closely the produced utterances matched observed child productions; indeed, Chang et al. (2006b) advocated this evaluation procedure as a cost-effective means of comparing models of syntax acquisition across a range of languages. In the limit of infinite evaluation data, this evaluation metric comes down to the test-set perplexity of the evaluation data (per-word or per-utterance). If a model does well on a perplexity metric, then it describes the distributional regularities of words well. However, syntactic knowledge is not about distributional regularities alone; it ultimately functions to relate the observed linear order of words to their unobserved semantic composition (or non-composition). It is possible that two models would obtain comparable perplexity scores, but one model succeeds by modelling unigram frequency and collocations well and the other one succeeds by modelling verb transitivity well. While the first model may be a good model for some aspects of linguistic knowledge (perhaps named entities or multi-word expressions), the second is clearly a better model of *syntactic* knowledge.

One alternative to a perplexity-like metric would be to evaluate the probability

distributions over syntactic representations in terms of their utility for a semantic composition system, a kind of "extrinsic" evaluation. Kwiatkowski et al. (2012) did just this, learning unobserved syntactic CCG derivations that relate observed word strings to observed logical forms. However, the evaluation would be sensitive to how we decided to map probability distributions over syntactic representations to logical forms, itself a non-trivial task, and annotations for this kind of evaluation are expensive in any case. For the evaluations in Chapters 5 and 6, we will evaluate parses against linguist intuitions about syntactic representations (i.e. hand-annotated parses) as a suitable middle-ground between a perplexity-like evaluation and a semantic composition-based extrinsic evaluation.

The next section presents the Dependency Model with Valence, which will form the core of the model used in Chapter 6.

3.3.2 The Dependency Model with Valence

The work in Chapter 6 will use a probabilistic model, called the Dependency Model with Valence (DMV), that explicitly computes probability distributions over potential dependency structures for observed word strings. This model makes reasonable assumptions about how different parts of a dependency tree covary, and then computes the probability distribution over dependency parses given those assumptions and a corpus of utterances. This section describes the historical development of these models in Natural Language Processing, and then provides an overview of the DMV with the intention of developing intuitions for why and how it works.

3.3.2.1 Historical context

It was recognized early on that learning a grammar from observations alone is difficult. Due to the uncertainty about grammatical structure inherent to the problem, most computational work on syntax acquisition has focused on statistical approaches.

As discussed in Section 3.2.3.1, syntax is usally analyzed in terms of a tree-like structure of hierarchically-embedded phrases. Such trees can be described with a Context Free Grammar (CFG), which summarizes all possible trees by listing rules that record the possible immediate children of each kind of phrase. For example, a CFG would include a rule $S \rightarrow NP\ VP$ to summarize that S nodes can immediately dominate an NP node followed by a VP node. Each possible sequence of immediate children is called the parents' expansion, and using a CFG implicitly assumes that the parent

can immediately dominate any of its expansions, regardless of what else is happening in the tree. We can make a CFG statistical by associating probabilities with each rule, such that the probabilities of each rule with the same parent sum to one. This is called a Probabilistic Context Free Grammar (PCFG), and strengthens the assumption that potential expansions are free of context into an assumption that the probability of each expansion is statistically independent of context, given the parent.

Baker (1979) introduced the Inside-Outside algorithm for training PCFGs. Inside-Outside is a version of the Expectation-Maximization algorithm, which in turn is an algorithm for finding parameters that make the observed data easy to expect. For example, if we see that a coin has come up heads eighty times in one hundred flips, this outcome is easy to expect if we assume the probability of heads for the coin is 0.8. The Inside-Outside algorithm finds a set of probabilities for PCFG rules that make observed word strings as likely as possible in the same sense. Early application of the Inside-Outside algorithm to learning PCFGs for natural language syntax indicated that vanilla PCFGs are too unconstrained to learn syntax, however. Pereira and Schabes (1992), for example, found that the Inside-Outside algorithm was not effective for discovering unlabeled constituency structure. Specifically, running the Inside-Outside algorithm on transcribed speech substantially increased the probability of the data, but did not improve parsing performance. Pereira and Schabes did obtain a large improvement, however, when they modified the Inside-Outside algorithm to be constrained by gold-standard unlabeled constituency annotations.

Constituency grammars introduce three significant sources of difficulty for unsupervised induction. First, a constituency grammar devotes a large number of parameters to modeling the validity of different linear orders, because each different possible sequence of immediate children gets its own probability. Second, since almost all of the parameters involve relations between unobserved phrasal nodes, syntactic events high up in a constituency tree bear only a very abstract relation to the observed words. Third, there are a huge number of possible binary trees even for short sentences.

Carroll and Charniak (1992) proposed dependency grammar as a grammar formalism that addresses the second and third point. As just discussed, dependency grammars produce analyses such as the one shown in Figure 3.6. The head of each dependency is at the tail of the arrow, and its dependent lies at the point of the arrow (Mel'čuk, 1988). If a dependency structure can be drawn without crossing any arrows, such as the one in Figure 3.6, it is a *projective* dependency structure. Projective dependency grammars can be represented using a context free grammar with a metavariable h to

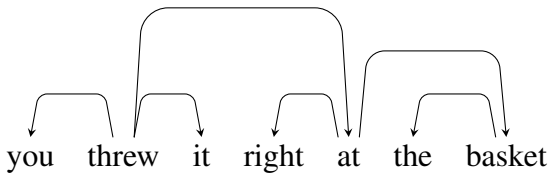


Figure 3.6: Example unlabeled dependency parse.

represent heads, and metavariables d_1, d_2, and so on, to represent dependents. Using these meta-variables, we can represent a dependency grammar using a PCFG that has context-free rules of the form $h \rightarrow d_1 \ \ldots \ d_i \ h \ d_{i+1} \ \ldots \ d_n$. That is, a head can expand to itself and its dependents. For example, to generate the tree in Figure 3.6, we would have a rule "threw $\rightarrow$ you threw it at." Because every arc is between two words, the number of possible trees is reduced: the number of arcs is just the number of words minus one. Second, because the words are always observed, and arcs are always between two words, the hidden structure is much more grounded in concrete observation. Unfortunately, Carroll and Charniak (1992) found in experiments on artificial data that this constraint was insufficient on its own.

Yuret (1998) and Paskin (2001) both pointed out that dependency grammar can also address the first source of error mentioned above; rather than modeling all the dependents at once, they proposed dependency parsing models that considered only one arc at a time. Specifically, rather than using context-free rules that generate all the dependents, their model used context-free rules that expand the head to itself plus one dependent: all rules were of the form $h \rightarrow h \ d$ or $h \rightarrow d \ h$. This kind of model is called an *arc-factored* model. The first rule is used when the dependent is to the right of the head, and the second is used when the dependent is to the left of the head. Yuret's model set parameters largely using heuristics, while Paskin's model contained an explicit PCFG formulation that is learned through EM. Unfortunately, both Yuret (1998) and Paskin (2001) underperformed a simple baseline that attaches a dependent to the adjacent word.

Klein and Manning (2004) provided the first unsupervised dependency parser to outperform the adjacent-attachment baseline. The basic insight was that Carroll and Charniak (1992) had constrained the grammar form too little by modeling every sequence of dependents separately, and that Paskin (2001) had constrained the model too much by not modeling dependent sequence at all beyond the arc direction. The explicit model formulation will be discussed in more detail shortly, but the fundamental idea

is that most syntactic regularities can be captured by modeling the direction of the attachment and the number of dependents a word has in that direction. The direction of the attachment matters because, for example, there is a strong preference for subjects to come before their verb, and objects to come after their verb. Similarly, the number of dependents in a direction matters because some words prefer very strongly to take only one or no dependents in particular directions. For example, intransitive verbs typically take a subject to their left, and no dependent to their right, while transitive verbs typically take a subject to their left and an object to their right. Alternatively, nouns prefer to take dependents, such as adjectives, to their left (e.g.: "big bad wolf") but not to their right. The theoretical syntax literature refers to the number of arguments a verb takes as its "valence," and so this model is called the Dependency Model with Valence (DMV).

Because of its success in outperforming the adjacent-attachment baseline, virtually all work on unsupervised parsing has been based on the DMV. Extensions have focused on improving the initialization procedure (Gimpel and Smith, 2012), proposing new learning regimes (Smith and Eisner, 2005; Spitkovsky et al., 2010), and providing Bayesian versions of the DMV (Cohen and Smith, 2008; Headden et al., 2009).

3.3.2.2 Overview of the DMV

For our probabilistic approach to the acquisition of syntactic dependencies in Chapter 6, we will be interested in computing a probability distribution over possible grammars θ (one grammar is a set of probabilities on all possible rules) and possible unobserved dependency trees $\mathbf{t}$, given a corpus of observed word sequences C and prior biases α: $P(\mathbf{t}, \theta | C, \alpha)$. We can understand this better by applying Bayes' rule:

$$P(\mathbf{t}, \theta | C, \alpha) = \frac{P(C, \mathbf{t} | \theta) P(\theta | \alpha)}{P(C)} \propto P(C, \mathbf{t} | \theta) P(\theta | \alpha) \tag{3.1}$$

$P(C, \mathbf{t} | \theta)$ is the probability of observed strings and unobserved dependency analyses under a particular grammar, and is called the likelihood. When the likelihood is high, the data is probable according to the grammar θ; intuitively, θ is a good explanation of the data. The DMV fundamentally is a likelihood function for dependency parses and word strings. As suggested by the numerator in the right-hand side of the equality in Equation 3.1, the DMV multiplied by the prior provides a probability distribution over all possible dependency parses for all possible utterances, including utterances of other languages. To obtain a probability distribution over trees $\mathbf{t}$ and parameters θ given the language we are working with, we divide the probability distribution for all trees of all

utterances by the probability of the utterances we actually saw (which appears in the denominator of the right-hand side of the equality). This section provides an intuitive outline of the DMV itself; Chapter 6 will provide a more technical discussion and describe how the DMV is incorporated into the other terms of Equation 3.1 in more detail.

Broadly, the DMV computes the likelihood of a dependency tree and words by noting that dependency trees for different sentences $s \in C$ will often share partial structures, such as an arc between a particular noun and the determiner that precedes it. However, the arcs themselves are unobserved, and instead we will build all possible dependency trees, and count how many times an arc appears in each possible dependency tree for each possible sentence, weighted by the probability of the trees containing that arc. Thus, a partial structure which potentially appears many times will be more probable than a partial structure which potentially appears very rarely. For example, the sequence "threw it" appears in many sentences, and so there are many opportunities for an arc between "threw" and "it." Our model will thus give relatively high probability to arcs between "threw" and "it." By contrast, the sequence "it right" is not very common, and so there are few opportunities for an arc between "it" and "right," and the model will likely allocate relatively little probability to arcs between "it" and "right."

The example of "threw it" is misleading in two respects, however. First, the model is not restricted to adjacent sequences, but assesses opportunities for arcs in terms of all possible projective dependency trees, so arcs between words that are never adjacent could still receive high probability. Second, the "number of opportunities" for an arc is assessed not only in terms of the number of sentences it appears in but also in terms of the probability of the trees in which it appears. Accordingly, if two words appear in many sentences together, but are linked by arcs only in trees that are, due to the other arcs of those trees, low-probability, an arc between those two words will still be low-probability. Conversely, if two words appear in only one sentence together but are connected by arcs in high-probability dependency parses, an arc between those words may well be high-probability.

More specifically, the DMV computes the probability of a dependency parse as, in part, the product of the probability of the individual arcs, meaning it is an *arc-factored* model.[4] Following the intuition mentioned above, the DMV tracks two aspects of par-

[4]The astute reader will notice a chicken-and-egg problem; the probability of an arc depends on the probability of the trees it appears in, but the probabilities of those trees depends on the probabilities of

tial structure. First, it tracks individual arcs: how often each word (potentially) took other possible words as dependents. Second, it tracks dependent counts: how often each word (potentially) took no dependents, one dependent, or more than one dependent. Chapter 6 will describe in detail how these substructures are hypothesized and counted, but the central idea is simple: the DMV makes syntactically-motivated assumptions about how different parts of syntactic trees covary. For example, it assumes that the probability that a specific potential head takes a specific potential dependent does not change if the potential head is itself a dependent, compared to a root. The probability that "threw" takes "it" as a dependent in Figure 3.6 would be the same even if the sentence were a subordinate clause of another sentence.

This assumption is a *conditional independence* assumption, and, as noted in Section 3.3.2.1, can be thought of as a statistical analogue to the Context Free assumption of Context Free Grammars. Using Context Free assumptions, variation in distributional behavior is captured by introducing new non-terminal symbols; concretely, to distinguish between transitive and ditransitive verb phrases, a modeler must introduce a transitive symbol and ditransitive symbol to the grammar, and then provide different possible expansions for each new symbol. Using conditional independence assumptions, variation in distributional behavior is captured in a similar way; concretely, the modeler may introduce new symbols for transitive and intransitive verbs or verb phrases, and specify different probability distributions over different expansions for these new symbols. However, conditional independence assumptions are more expressive, because modelers can also specify different probability distributions over the *same* expansions. Such expressivity is useful for lexicalized grammars, because it allows the grammar to represent, e.g., that different verbs may be either transitive or intransitive, but some verbs are biased to be transitive and others are biased to be intransitive. Chapter 6 will show that the DMV adds symbols to the grammar in a way that captures precisely these kinds of biases.

In summary, a statistical approach to syntax acquisition focuses on the probability distribution over syntax trees, given the utterances a child hears. The probability distribution over syntax trees in turn depends on, in addition to what the child hears, assumptions about how different parts of the syntax tree co-vary (encoded in the likelihood function), and also on a prior bias over grammar parameters of a given form (encoded in the prior). The likelihood function is the more interesting component of

their arcs. Chapter 6 will circumvent this problem with a gradient-ascent algorithm called Mean-Field Variational Bayes. Other approaches use sampling algorithms.

this mix, and the DMV provides an empirically-verified and syntactically-motivated starting point for this likelihood function.

3.4 Conclusion

This chapter discussed bootstrapping accounts for language acquisition, and emphasized the role of dependencies between different kinds of linguistic knowledge in such accounts. Section 3.2 showed how this leads to a dimensionality reduction view of bootstrapping accounts: if they are fundamentally about exploiting dependencies between different kinds of knowledge, then bootstrapping strategies work because they situate grammars in a smaller space that is easier to represent and explore. Section 3.3 proceeded to show how the dimensionality-reduction view motivates computational modeling as an important complementary source of evidence about language acquisition. Specifically, different kinds of grammatical knowledge "live" in complex spaces with discrete and continuous components, and so the dependencies underlying realistic bootstrapping accounts will have complex functional forms. While laboratory experiments can determine whether children respond to relevant covariates in the expected way, computational modelling approaches can measure the practical utility of these dependencies in the evidence children have.

Chapter 4

Predictability Effects in Child-directed Speech

4.1 Introduction

This dissertation explores the possibility that effects of language predictability on word duration are useful for a language-learning infant. One important prerequisite to demonstrating this possibility is showing that predictability effects exist in child-directed speech. In this chapter, mixed effects regressions on adult-directed and child-directed speech will show that predictability effects do exist in child-directed speech, and that at least some of them are indistinguishable in direction and magnitude from the predictability effects of adult-directed speech. This result will bolster our confidence that children may use such effects.

As a secondary line of inquiry, we will also employ this regression methodology to explore whether, and how, talkers pursue different redundancy strategies in response to changes in listener characteristics and communicative constraints. As introduced in Chapter 2, it has been proposed that predictability effects make speech more efficient by providing a shorter signal when there is a lower probability of error. Previous work, however, has mostly shown that predictability effects exist, without explicitly tying them to communicative efficiency. Our first set of regressions will show that talkers produce different predictability effects when addressing prelinguistic children compared to adults, providing evidence that talkers adjust predictability effects according to communicative factors.

A second set of regressions will examine how predictability effects are affected by listener visibility. Talkers addressing visible listeners presumably enjoy a higher chan-

nel capacity than talkers addressing non-visible listeners, and so, assuming a goal of efficiency, should produce less redundancy. Our regressions find that talkers addressing visible listeners do produce less redundant speech than talkers addressing non-visible listeners, suggesting that talkers actively exploit the extra channel capacity afforded by visual cues.

Together, these results will provide evidence of a close tie between effects of predictability on word duration and communicative demands: talkers adjust predictability effects in response to listener characteristics, suggesting some degree of audience design, and talkers reduce redundancy as the channel capacity increases, indicating that talkers aggressively adapt to communicative demands on redundancy. We will now discuss previous results regarding listener and channel characteristics and predictability effects, before examining predictability effects on two corpora of adult-directed speech and one corpus of child-directed speech.

4.2 Background

Chapter 2 discussed how predictability effects lead to a more efficient communication system by reducing signal redundancy when the probability of an error is very low. That chapter closed by pointing out that results so far do not show that predictability effects improve the efficiency of communication very much. Thus, it is still possible that predictability effects have little to do with communicative efficiency, in either cause or effect. The results from this chapter will more tightly link predictability effects and communication.

In order to understand how predictability effects might exist without impacting communication very much, consider the following possibility. These effects could be artifacts of the way that lexical and grammatical knowledge is stored and accessed. It has been proposed (e.g. TRACE McClelland and Elman, 1986) that lexical entries are accessed by way of a spreading activation mechanism with competition. In these models, each node in an undirected graph corresponds to a word, and when a node's activation level reaches a threshold, the word is retrieved. To prevent multiple "winners," a node inhibits the activation of its neighbors; as a node's activation increases, it inhibits its neighbors more strongly. Simple modifications to this kind of network could result in predictability effects; perhaps frequent words may inhibit their neighbors more strongly or have a higher resting activation (Dahan et al., 2001). Such modifications could conceivably be an accident of biology; frequently activated neural

connections tend to be stronger (Bliss and Gardner-Medwin, 1973; Bliss and Lømo, 1973), for example, which could lead to a more robust inhibition for frequent words. In this situation, predictability effects could arise without any pressure from communicative efficiency, and perhaps without affecting the efficiency of communication very much.

Accordingly, both the communicative efficiency view and accident-of-biology view are consistent with the existence of predictability effects. These views differ, however, in how predictability should be computed. In Chapter 2, we noted that an optimal redundancy scheme must be well-adapted to the particular noisy channel. For speech, this means that talkers can become more optimal if they attend to listener and channel characteristics in assessing how predictable a word is. Under the accident-of-biology view, by contrast, it is unlikely that listener and channel characteristics would matter. Thus, the efficiency view predicts that talkers modulate predictability effects according to listener and channel characteristics, and the accident-of-biology view does not. We discuss existing evidence regarding these predictions next.

4.2.1 Listener characteristics

Bard et al. (2000) examined whether talkers adjusted to listener knowledge in a spontaneous speech task. In this task, the "Map Task," they recorded several pairs of people negotiating a route on a map. One member of each pair, the 'guide', had a map with a route drawn on it, and the other, the 'follower,' had a similar map without the route. The participants talked with each other to draw the same route on the follower's map. Crucially, some of the landmarks on the guide's map were missing from the follower's map.

These missing landmarks are clearly less predictable for the follower, and so, under the efficiency view of predictability effects, we would expect a guide to reduce a missing landmark less heavily once they know that landmark is missing on the follower's map. However, Bard et al. did not find any evidence that talkers considered listener knowledge in reduction. Specifically, talkers reduced landmarks once either participant had introduced the landmark into the conversation, regardless of *who* introduced the landmark. More importantly, they found that talkers continued to reduce the landmark even if the partner was changed. As this new partner could not possibly know about a landmark that was missing on their own map, this result provides strong evidence that talkers were not tracking predictability from the listeners' point of view–they were

at most simply using their own knowledge as a proxy for their partners' knowledge. Bard et al. thus concluded that predictability effects are largely driven by automatic, talker-centric processes.

While Bard et al.'s study suggests that talkers do not maintain detailed models of listeners' discourse knowledge to determine word predictability (and thus pronunciation), it is still possible that talkers modulate predictability effects for the benefit of the listener in response to more general knowledge about listener characteristics, such as the listener's overall linguistic competence, or communicative goals. In short, talkers may consider listener characteristics and listener knowledge that are easy to assess and track, but fall back on a talker-centric model as these listener characteristics become more difficult to compute.

Our first set of regressions will provide evidence in favor of this weaker version of listener-based predictability effects, showing that the predictability effects of child-directed speech are different from those of adult-directed speech.

4.2.2 Channel characteristics

As discussed in Chapter 2, as a noisy channel's capacity increases, a code can safely incorporate less redundancy without increasing the error rate. This is because the capacity of a noisy channel approaches the entropy of the source as the channel becomes more reliable, and the entropy of the source includes zero bits for redundancy. Under the efficiency view of predictability effects, then, we would expect talkers to exhibit less redundancy in a channel with a higher capacity.

Boyle et al. (1994) provided early evidence that speaks to this question by looking at the behavior of the same talkers using the same data as Bard et al. (2000). Specifically, in that data, some pairs of speakers could not see each other during the task, and some pairs could. Presumably, pairs in the Visible condition enjoyed a higher channel capacity than pairs in the Non-Visible condition. This intuition is supported by the fact that pairs in the Visible condition completed the task more quickly than pairs in the Non-Visible condition, conveying the same amount of information (the map route) in less time. Boyle et al. found that pairs in the Visible condition sought to establish eye contact more often during periods of communicative difficulty, providing some evidence that visibility increases channel capacity in a way that speakers seek to exploit.

Our second set of regressions, using the same data as Boyle et al. (1994) and Bard

et al. (2000), will show that the Map Task talkers exhibited fewer predictability effects in the Visible condition. This result indicates that talkers attend to channel characteristics in the expected way when modulating reduction. Additionally, this second set of regressions will serve as a control for the first set, which looked for differences between predictability effects in CDS and ADS. The CDS corpus was gathered such that the mothers and children were visible to each other, but the ADS corpus is composed of telephone conversations with Non-Visible interlocutors. By looking at the Map Task data, we will be able to isolate the effect of visibility, and provide evidence that the differences in the predictability effects of CDS and ADS cannot be attributed simply to an effect of visibility.

4.3 Experiment I – child- and adult-directed speech.

In this experiment, we will look at predictability effects in child- and adult-directed speech, first to verify that predictability effects exist in child-directed speech, and second to see if coarse listener characteristics influence predictability effects. Our general methodology is borrowed from the corpus study of Bell et al. (2009) investigating predictability effects. The current study differs from Bell et al. (2009) in comparing predictability effects in ADS and CDS, using mixed effects regression rather than standard regression, and residualizing against a control model to minimize collinearity between predictors of interest and control predictors.

4.3.1 Data

We use data from two corpora: `swbdnxt` (Calhoun et al., 2010), an edition of the Switchboard corpus of telephone conversations between adults, and `Large Brent` (Rytting et al., 2010), a subset of the Brent corpus (Brent and Siskind, 2001) of spontanous child-directed speech. We describe these corpora and the data extracted from them below.

4.3.1.1 `swbdnxt`

`swbdnxt` is an edition of the Switchboard corpus of telephone dialogues between adults. It integrates several levels of annotation produced by different groups since the original Switchboard release. These include prosodic (ToBI) and syntactic annotations, as well as a phonetic alignment created by correcting the output of a forced

Table 4.1: Statistics for the two datasets used in Experiment I.

	# Sent	# Words	$\frac{\text{Word}}{\text{Sent.}}$	# Talkers
`swbdnxt`	2,273	16,301	7.2	73
`brent`	2,254	14,148	6.3	4

alignment produced using a pronunciation dictionary.

To create the dataset for our experiments, we began with the 75 conversations that are annotated with ToBI, Mississippi state phonetic alignments, and Penn Treebank POS tags and parses. We discarded one conversation due to inconsistent annotation, and one talker due to missing metadata. The resulting corpus contained $12,140$ sentences ($99,965$ words), from which we removed all sentences longer than 20 words and shorter than 3 words, eliminating $4,704$ sentences ($39,452$ words). Long sentences were excluded so that the range of ADS sentence lengths would be approximately the same as the range of CDS sentence lengths. Short sentences were excluded in anticipation of the exclusion, described shortly, of short prosodic phrases.

We split the remaining sentences into 80% training, 10% development, and 10% test sets in anticipation of the experiments of Chapter 5 and 6. For this study, we use only the training set, and only talkers of side A (to avoid having to handle correlations between talkers in the same conversation), a total of 23,638 words from 2,608 sentences. Following Bell et al., we discard all words adjacent to disfluencies (identified by the POS tags "UH" and "XX") to avoid the complicated effects of disfluencies on word duration. Also following Bell et al. (2009), who note that short prosodic phrases are typically formulaic discourse responses (i.e. "oh good grief"), we discard prosodic phrases that are three words or shorter. Table 4.1 shows statistics for the final `swbdnxt` corpus.

4.3.1.2 `Large Brent`

`Large Brent` is a subset of the Brent Corpus of spontaneous CDS collected in a naturalistic setting. It consists of the mothers' utterances from four mother-infant dyads, and has a forced phone alignment based on a modified version of the CMU pronunciation dictionary. Details of the corpus and alignment can be found in Rytting et al. (2010). `Large Brent` has a 90%/10% train/test partition; for this study we use only the training partition, which contains $22,226$ words from $7,030$ sentences. Rytting

et al. have already excluded utterances containing partial or unintelligible words, so we made no further effort to handle disfluencies.

Unlike `swbdnxt`, this corpus does not include talker's ages; since all talkers are new mothers, we use an estimate (based on personal communication with Michael Brent) of 27 years old for all talkers. There is also no annotation of intonational phrase boundaries, which are known to affect word duration. However, in this corpus every pause of 300 milliseconds or more is taken to be an utterance boundary, so we use the utterance boundaries as a fairly robust approximation to intonational phrase boundaries. As in the ADS corpus, we remove all prosodic phrases which are three words or shorter, resulting in the corpus statistics shown in Table 4.1.

4.3.1.3 Pooled dataset

To facilitate direct comparison between predictability effects in ADS and CDS, we created a pooled dataset containing the data from both `swbdnxt` and `Large Brent`. As can be seen in Table 4.1, the pooled dataset is relatively balanced in terms of the number of sentences and words from each type of speech, but not in terms of talkers. The imbalance in the number of talkers is handled by including a random effect for Talker in our model (described below).

4.3.2 Models

4.3.2.1 Approach

Word duration is affected by many factors other than word predictability, such as talker age, speech rate, the word's length in phones, and its position in the intonational phrase. To control for these kinds of factors, we adopt a simple two-step regression procedure. First, we build a single control model (using the model selection procedure described shortly), which regresses log word duration against only control terms such as those above.[1] The control model is fitted to the pooled dataset, and includes a Speech Type term (ADS or CDS) to allow for effects not related to predictability on duration due to speech type. We also allow interactions between Speech Type and the other control factors. Next, we take the residuals of this control model as the response variable for model selection among predictability terms, so that these terms can only be used to explain the part of the variance that has not already been accounted for by control

[1] Like Bell et al., we take the log of the duration to avoid equating a 50ms difference in a word that is usually 60ms with a 50ms difference in a word that is usually 300ms.

factors. We perform three separate regressions on the residuals: one on the residuals from the ADS subset of the data, one on the CDS subset, and one on the entire pooled dataset. The ADS and CDS regressions allow us to assess which predictability terms are significant factors in predicting word duration in ADS or CDS, while the pooled regression can show, through interactions with the Speech Type term, whether a given predictability factor has different effects in ADS and CDS.

This two-step approach has the advantage that we do not need to worry about collinearity among our control predictors. If two predictors are collinear, the parameters of the terms in the control model will be unstable, but the overall variation explained, and the residuals, will be stable.

The approach outlined above could be used with a standard linear regression model. However, such a model assumes that data points are sampled independently, which is clearly not true for words from the same sentence or from the same talker. Instead, we use a mixed effects regression, which generalizes standard regression by including multiple random effects rather than only one (the error term). Specifically, we will be able to estimate different random baselines and slopes for each talker and for each sentence. The random effects will not be examined directly; rather, they will "soak up" the otherwise unhandled correlations between items, providing us with more robust parameter estimates without sacrificing statistical power. We discuss the details of the random effects and model selection procedure below, after describing the fixed effects used in our control and predictability models.

4.3.2.2 Fixed Effects: Control terms

We include nine terms in the control model, based on those of Bell et al. (2009). These are: **Talker Age** (taken from metadata in `swbdnxt`; estimated as described in the Data section for `Large Brent`), **Talker Sex**, **Speech Rate** (computed per utterance as $\log\left(\frac{\#\text{ Vowels}}{\text{Second}}\right)$), **# Vowels** (taken from annotation based on a pronunciation dictionary), **Average Word Duration** (computed as the sum of the average duration of each phone in the word, following Bell et al.; average phone durations were computed separately for each dataset), **Intonational Phrase Initial** (indicates whether a word is at the beginning of an intonational phrase; phrases are bounded by break indices of 3 or 4 in `swbdnxt` and are assumed to coincide with utterances in `Large Brent`), **Intonational Phrase Final** (as previous), **Content or Function Word**[2] (based on POS

[2] Bell et al. investigate this term in interaction with predictability terms. We attempted to include it in our predictability model, but it is highly collinear with other predictability terms and we failed to

tags), and **Speech Type** (ADS or CDS).

4.3.2.3 Fixed Effects: Predictability terms

We include four predictability terms, again following Bell et al.: **Log Word Frequency**, **Preceding Context** (log probability of a word given the preceding two words), **Following Context** (log probability of a word given the following two words), and **First Mention** (whether or not the talker has said the word). All of these terms are computed from the conjunction of the CDS and ADS corpora prior to discarding short prosodic phrases. We also tried computing these terms on the final dataset, after discarding short prosodic phrases, and found no significant differences in the modeling results. The first three predictors are all Good-Turing smoothed. To reduce collinearity, we residualized Word Frequency against the other three predictors.

4.3.3 Model Selection

We employ a model selection procedure that closely follows an algorithm introduced by Coco and Keller, with only minor modifications to avoid specific interactions that lead to convergence errors (all our CDS talkers are female, for example, so we avoid testing for an interaction between Speech Type and Sex).[3] For each round of model selection, we consider two random effects: Talker, and Sentence. We first determine which of the two random effects (intercept only) produces a better initial model. We then determine whether adding the other random effect (intercept only) produces a significant improvement in model fit. This is followed by another series of model comparisons to add fixed main effects and random slopes for each random effect (including random effects that are not already in the model). Finally, we perform another series of model comparisons to add interactions.

In each step, a predictor is added if the model with that predictor is a significantly better fit to the data than the model without that predictor, as assessed with the `anova` function for model comparisons in `R`. The current implementation of the model selection procedure requires every model form under consideration to have at least one random intercept. As the control model contains both random intercepts, residuals from the control model will exhibit no variation for the random intercepts and lead

reduce collinearity to an interpretable level, so we include it in the control model instead.

[3]The R implementation of the modified algorithm is available at http://homepages.inf.ed.ac.uk/s0930006/modelselect.R.

to convergence problems when performing model selection among predictability predictors. To ensure that there is variation in the predictability models for a random intercept by Sentence or Talker to explain, we first remove the random intercept by Sentence and Talker from the control model formula before obtaining residuals for the predictability models. In the results to be presented, we explored models that assumed random slopes and intercepts were conditionally independent, given the random variable. These models also did not estimate random slopes for interactions because models with random slopes for interactions almost never converge. We also explored some of the other possibilities (random slopes for interactions, correlated slopes and intercepts), but consistently obtained the same patterns of results.

As the model selection procedure involves several dozen model comparisons, we are conservative in assessing the significance of model comparisons. Specifically, whereas Coco and Keller consider a model comparison significant if the larger model is a better fit at $p < 0.05$, we require $p < 0.01$. All predictors were centered except Speech Type; for ease of interpretation, Speech Type was set to -1 for adult-directed Speech and 1 for child-directed speech, resulting in a mean value of ≈ -0.071.

While mixed effects models provide a t-statistic for each fixed effect, how to obtain a p-value from this t-statistic is controversial. One way to obtain p-values is to follow heuristics, based on the number of observations and number of model parameters, for picking a degree of freedom for the null-hypothesis t-distribution. We are working with relatively large datasets, so commonly-used heuristics will end up picking degrees of freedom so large that the resulting t-distribution is indistinguishable, in practice, from a z-distribution (i.e. a standard normal distribution). Accordingly, we will treat our t-values as z-values, and obtain two-sided p-values. Barr et al. (2012) found, using simulated data, that this method achieves relatively good Type I error rates and power.[4] To emphasize the inexactness of this method, the tables presented will contain only t-values; as a rule of thumb, a t-value with an absolute value greater than 2 is significant at a critical level of 0.05, and a t-value with an absolute value greater than 2.6 is significant at a critical level of 0.01.

[4] Barr et al. (2012) also found that using model comparison achieves good Type I error rates and power, which is the method we use during model selection. Additionally, they report that using a Markov Chain Monte Carlo (MCMC) approach achieved very high Type I error rates, and preliminary experiments with more complex simulated datasets indicate that Type I error rates increase dramatically as the complexity of the dataset increases. Some of the results of this chapter were originally presented in Pate and Goldwater (2011a) using the MCMC method, and we will see that those results stand when using this more rigorous method.

(a) Coefficients of the individual ADS model.

Predictability Term	Coeff.	*t*-val
(Intercept)	-0.0198	-3.73
Word Freq.	-0.0205	-10.36
Prec. Context	-0.1296	-11.98
Foll. Context	-0.0336	-2.53
First Mention	0.0473	6.34

(b) Coefficients of the individual CDS model.

Predictability Term	Coeff.	*t*-val
(Intercept)	-0.0176	-4.51
Word Freq.	-0.0231	-11.26
Prec. Context	N/A	N/A
Foll. Context	-0.0727	-7.55
First Mention	0.0471	6.38

4.3.4 Results

4.3.4.1 ADS and CDS individually

Before performing a direct comparison of ADS and CDS data, we first build individual predictability models for the two types of speech. The individual models serve two purposes. First, they tell us what kinds of effects are present in each kind of speech, allowing an informal comparison of the predictability effects in ADS and CDS speech. Second, we can use the results of these individual models to inform model selection when performing a direct comparison on the pooled data. In short, the individual models identify patterns of significant effects, and the pooled model compares these patterns of significant effects in a quantitative manner.

For the individual models, we take the residuals from the control model fitted on the entire pooled dataset, and run model search on the CDS predictability terms to predict the residuals for CDS words, and then run model search on the ADS predictability terms to predict the residuals for ADS words. We do not consider interactions, as they are difficult to interpret.

Results for ADS and CDS are presented in Tables 4.1(a) and 4.1(b), respectively. As the response variable is (residual) log duration, a negative coefficient corresponds to a shortening effect as the predictor increases, and a positive coefficient corresponds to a lengthening effect. For both ADS and CDS, a number of predictability effects are observed. For ADS, we find all the expected predictability effects among the main effects: frequent words and words predictable from context are shorter, while words new to the conversation are longer. Accordingly, we have replicated previous findings on ADS. In CDS, we also find effects, in the expected directions, of Word Frequency, First Mention, and Following Context, but Preceding Context was not added to the model (and it is not significant if its addition is forced). We discuss the implications of

this difference in Section 4.3.5.

These individual models, however, only reveal whether we have enough evidence to determine that the various coefficients are significantly different from zero, without comparing the effects in ADS with those in CDS. A direct comparison accomplishes two goals. First, it can confirm that the effect of Preceding Context is actually weaker in CDS than it is in ADS. Second, it is possible that an effect might be significant and in the same direction in both types of speech, but be much stronger in one type of speech. A pooled model on the entire dataset enables just such a direct comparison of effects in each Speech Type by examining interactions with the Speech Type term. We now proceed to this pooled model.

4.3.4.2 Pooled comparison

In this section, we perform model search on the full pooled dataset, and include in the predictability model the fixed effect "Speech Type" that indicates whether each word is from the ADS dataset or the CDS dataset. We will in particular be examining the interaction terms between Speech Type and the predictability terms. Since we wish to verify different patterns of significant results in models containing only main effects, we perform model search up to 2-way interactions. Model search proceeds as before, using residuals taken from the same pooled control model as the response variable, except we force the addition of 2-way interactions between Speech Type and First Mention.[5]

Table 4.2 presents the model coefficients and t-values from our final model. The table is separated into three boxes for each order of interaction, and each box is split into interactions not involving Speech Type on the top and interactions involving Speech Type on the bottom. There is relatively little collinearity in this model; most of the pairwise correlations among main effects are less than 0.1 in magnitude, with only Speech Type and First Mention above 0.2 (but below 0.3). When looking to interaction terms, there is a correlation of -0.6 between Preceding Context and its interaction with First Mention, and also a correlaton of -0.8 between Word Frequency and its interaction with First Mention. When we forbade these interactions during model search, the t-values of the other terms changed only negligibly, and the same pattern of significant differences persisted. In addition to these collinear interactions, one correlation involving interaction terms was between 0.3 and 0.4, one was between 0.2 and 0.3,

[5]We also tried forcing the addition of interactions between Speech Type and any term which was added to the individual models, but the additional interactions were not significant.

	Predictability Term			Coeff.	t-val
Main effects	(Intercept)			-0.0185	-5.66
	Word Freq.			-0.0114	-5.09
	Prec. Context			-0.0916	-9.65
	Foll. Context			-0.0380	-3.22
	First Mention			0.0479	9.03
	Speech Type			0.0023	1.15
Interactions	Word Freq.	×	First Mention	-0.0188	-6.13
	First Mention	×	Prec. Context	0.0616	4.03
	Word Freq.	×	Prec. Context	-0.0128	-3.133
	Word Freq.	×	Foll. Context	-0.0115	-2.85
	Speech Type	×	Prec. Context	0.0710	9.41
	Speech Type	×	Word Freq.	-0.0045	-2.92

Table 4.2: Fixed effects, Speech Type coded with CDS as 1.

and the rest were below 0.2, with the vast majority below 0.1.

As our Speech Type variable is approximately centered, the main effects terms indicate an approximate average over the entire pooled corpus (with a slight bias to ADS). We observe in the main effects the same predictability effects as we saw in the ADS individual model: a significant and lengthening effect of First Mention, and a significant and shortening effect of Word Frequency and contextual probabilities.

Looking to the interactions that involve Speech Type, we first see a significant and positive interaction between Speech Type and Preceding Context, indicating that the shortening effect of Preceding Context is weaker in CDS. Moreover, since the interaction coefficient is roughly equal in magnitude to the main effect coefficient, this confirms the individual model findings that there is little or no effect in CDS. Secondly, we find a significant negative interaction between Word Frequency and Speech Type, indicating a stronger shortening effect of Word Frequency in CDS.

4.3.5 Discussion

These results show many similarities in the effects of predictability on word duration in CDS and ADS. Both speech types exhibit significant lengthening for First Mentions, and significant shortening according to Frequency and Following Context. Thus, the

main goal of this chapter is accomplished: predictability effects are clearly available to children.

In considering the possibility that predictability effects relate to communicative efficiency, however, our results did reveal some differences between ADS and CDS. Specifically, while Preceding Context showed a significant shortening effect in ADS, it appeared to have no effect in CDS. Additionally, although Word Frequency had a shortening effect in both speech types, we found a significantly stronger effect in CDS. These results indicate that these effects are, at least in part, modulated in response to listener characteristics. Our results cannot be explained under an information-theoretic approach by simply assuming a noisier channel in CDS, with talkers slowing down the overall rate of communication. Under this assumption, we would expect the same factors to be significant in predicting duration for both CDS and ADS. The coefficients might be different, but they would all change in the same direction. Instead we found a heightened effect of Word Frequency in CDS, and a diminished effect of Preceding Context.

Our results also raise an interesting question: what could explain the particular pattern of differences we found? Although a fully satisfactory account requires further investigation, we speculate that the factors found to be significant in both ADS and CDS are those that reflect talker-based mechanisms, while the remaining factor (Preceding Context) reflects accommodation to the listener. Our reasoning is as follows. First, there is a well-attested relationship between word frequency and the talker-based process of lexical access (Griffin and Bock, 1998, and references therein); this relationship is usually explained in terms of the resting activation of particular lexical items. Similarly, talker-based lexical access processes can account for the effect of First Mention: previously mentioned words have higher activation due to priming. The effect of Following Context can be explained using a different talker-based mechanism, in this case sentence planning. This explanation arises from spreading activation models of sentence production (e.g. Dell, 1986; Tily et al., 2009), which tie contextual probabilities to syntactic form activation.

Although the effect of Preceding Context may also be related to lexical access, we note that it is the only measure which always involves words the talker has just said (and the listener has just heard). Accordingly, it is the measure best-situated to capture tacit awareness of the listener's processing load, and so may be the only measure which captures primarily listener-based influences on word duration. If this is the case, then it may be that talkers have no control over the first three effects (as they are hard-wired

side effects of how language production works), but can control the effect of Preceding Context. Since infant listeners have little or no linguistic competence, talkers may simply "turn off" the effect of Preceding Context in CDS, knowing that the listener will be unable to make correct predictions about words in context.

In Experiment II, we will examine a manipulation of the channel itself by looking at data that varies whether speakers can see each other. This second experiment will serve two purposes. First, it will allow us to test the effect of increased channel capacity on predictability effects in a fairly direct way. Second, it will allow us to see if the results of Experiment I can be attributed solely to an effect of visibility rather than listener characteristics; the `swbdnxt` dataset consists of phone dialogues in which the interlocutors cannot see each other, while the mothers of the `Large Brent` dataset can see their children as they care for them.

4.4 Experiment II – The effect of speaker visibility

Experiment I showed that predictability effects exist in child-directed speech, but that they are somewhat different from the predictability effects of adult-directed speech. This result indicates primarily that predictability effects are available to children, and secondarily that talkers adjust predictability effects according to listener characteristics.

In Experiment II, we will see whether, and how, predictability effects vary according to visibility for two reasons. First, Experiment I did not control for a potential effect of visibility: all CDS talkers could see their listener, while no ADS talkers could. By looking at another dataset that manipulates only visibility, we will be able to see if the differences between ADS and CDS can be attributed to a difference in visibility *alone*. Second, as discussed in Section 4.2, communication with a visible partner presumably increases the capacity of the communicative channel. If predictability effects really are about providing extra redundancy to avoid errors in an efficient way, we would expect predictability effects to weaken as the channel capacity increases.

4.4.1 Data

For this experiment, we use data from the same HCRC `maptask` corpus used by Boyle et al. (1994) and Bard et al. (2000). This dataset is a collection of transcribed recordings of unscripted, task-oriented dialogue. Each dialogue has two participants, and

Table 4.3: Statistics for the Map Task dataset used in Experiment II.

	# Sent	# Words	$\frac{\text{Word}}{\text{Sent.}}$	# Talkers
`maptask`	7,592	77,485	10.2	64

they are both given a cartoon map with several labeled landmarks. One participant, the "guide," is given a map with a route drawn on it, while the other participant, the "follower," is given a map with no route. The guide and follower then converse to help the follower draw the route on his or her own map. In half of the dialogues, the guide and the follower can see each other, while in the other half they cannot. In neither condition did they see each other's maps. This dataset contains hand-annotated start and end times for each word.

To prepare the corpus for our study, we started with all dialogues forming an initial corpus with $20,719$ sentences ($145,483$ words). We then discarded sentences with fewer than four words, producing a dataset with $12,025$ sentences ($132,076$ words). Next, we discarded any word that had been annotated as part of a disfluency, producing a dataset with $11,782$ sentences ($107,759$ words). Finally, we discarded all follower utterances, leading to the dataset statistics in Table 4.3.

4.4.2 Models

We used the same basic modeling approach as in Experiment I, using the same model selection procedure to first fit a control model, and then performing model selection to find a regression of the residuals of the control model against the predictability factors. The set of control predictors for Experiment II were essentially the same as the control predictors for Experiment I. The corpus annotation provides the age of each talker, so we used the real age of the talker. Only a small part of the `maptask` corpus has ToBI-style break index labels, so, as with the `Large Brent` corpus, we assumed that utterance-final (-initial) words were the only prosodic phrase-final (-initial) words. Experiment II involved more random effects. The model selection function considered random intercepts and slopes according to sentence, talker, conversation, map, and talker birthplace.

(a) Coefficients of the Invisible model.

Predictability Term	Coeff.	t-val
(Intercept)	0.0089	2.93
Word Freq.	-0.0106	-4.18
Prec. Context	-0.0175	-14.36
Foll. Context	-0.0446	-34.68
First Mention	0.0088	2.97

(b) Coefficients of the Visible model.

Predictability Term	Coeff.	t-val
(Intercept)	-0.0002	-0.64
Word Freq.	0.0000	0.03
Prec. Context	-0.0019	-14.24
Foll. Context	-0.0041	-36.09
First Mention	N/A	N/A

4.4.3 Results

As in Experiment I, we first build individual predictability models for the Visible condition and the Non-Visible condition. These individual models will provide an intuition for which effects are and are not present in each Visibility condition. Similarly, to verify whether differences in patterns of significant effects are real, we will fit a full model on both conditions, and see if interactions with Visibility are significant.

4.4.3.1 Visible and Non-Visible Individually

Results for the Non-Visible and Visible conditions are presented in Tables 4.3(a) and 4.3(b), respectively. As before, the response variable is (residual) log duration, and a negative coefficient corresponds to a shortening effect as the predictor increases. Among the Non-Visible results of Table 4.3(a), we see the expected shortening effects of Word Frequency, Preceding Context, and Following Context, along with the expected lengthening effect of First Mention. However, looking to the Visibility results of Table 4.3(b), we see shortening effects only of Preceding and Following context; there is no apparent effect of Word Frequency, and First Mention was not even added to the model.

We turn next to a pooled comparison to see if the different patterns of significant results are significantly different.

4.4.3.2 Pooled comparison

Similarly to Experiment I, we fit the pooled model on the concatenation of the Visible and Non-Visible datasets. To see if the different patterns of significant results are different, we forced interactions between Visibility and First Mention and Word Frequency.

Table 4.4 presents coefficients and t-values from our final model. Among the main

	Predictability Term			Coeff.	t-val
Main effects	(Intercept)			0.0150	6.47
	Word Freq.			0.0026	1.43
	Prec. Context			-0.0161	-16.43
	Foll. Context			-0.0421	-46.81
	First Mention			0.0083	3.75
	Visibility			-0.0059	-2.71
Interactions	Word Freq.	×	Prec. Context	-0.0070	-12.68
	Word Freq.	×	Foll. Context	-0.0136	-32.82
	First Mention	×	Prec. Context	0.0044	6.06
	First Mention	×	Foll. Context	0.0108	12.72
	Prec. Context	×	Foll. Context	-0.0050	-22.17
	Visibility	×	Word Freq.	0.0043	2.38
	Visibility	×	First Mention	-0.0057	-2.81

Table 4.4: Fixed effects, visibility coded with Visible as 1.

effects, representing an average across the Visible and Non-Visible data, we see significant shortening effects of Preceding and Following Context, and a significant lengthening effect of First Mention. We do not see evidence of an effect of Word Frequency, averaged across the conditions. We also see a significant negative effect of Visibility, indicating that words in the Visible condition are overall shorter than words in the Non-Visible condition. This overall difference itself directly suggests that talkers pursue a higher communication rate with visible partners.

Among interactions, we see a significant ($p \leq 0.05$) positive interaction between Visibility and Word Frequency, indicating that any shortening effect of Word Frequency is significantly weaker in the Visible condition, and stronger in the Non-Visible condition. This result validates the Individual models' apparent result that Word Frequency has a shortening effect in the Non-Visible condition, but not in the Visible condition. We also see a significant negative interaction between Visibility and First Mention, indicating that any lengthening effect of First Mention is significantly weaker in the Visible condition, and stronger in the Non-Visible condition. This result validates the Individual models' apparent result that First Mention has a lengthening effect in the Non-Visible condition, but not in the Visible condition.

4.4.4 Discussion

These results show striking differences between reduction in the Visible and Non-Visible cases. The Non-Visible data showed shortening effects of Word Frequency, Preceding and Following Context, along with a lengthening effect of First Mention, all as expected, but the Visible data showed only shortening effects of Preceding and Following Context. This result first indicates that the difference we found between ADS and CDS in Experiment I probably cannot be attributed to an effect of Visibility. The effect of Word Frequency significantly strengthened in Visble CDS compared to Non-Visible ADS, but the effect of Word Frequency seems to disappear in Visible ADS; the effect of Preceding Context disappeared in Visible CDS, but remained about the same in Visible ADS compared to Non-Visible ADS; and the effect of First Mention disappeared in Visible ADS, but was readily apparent in CDS.

These results also indicate that these talkers provided less redundancy in the Visible condition than in the Non-Visible condition. In the Visible condition, we saw no apparent effect of Word Frequency or First Mention, and we also found that talkers in the Visible condition produced shorter words overall. Together, these two results indicate that talkers who could see their partner stopped adding redundancy (or added non-detectably small redundancy) to words that were rare or First Mentions.

Talkers still added about the same amount of redundancy according to Preceding and Following Context, however. Ignoring the different effects of First Mention for the moment, we can explain this pattern, in part, by appealing to the kind of control model we used. One of the primary goals of the control model is to remove variation that can be attributed to the basic word form. If lexicons are optimized, in terms of word length, for the usual communicative scenario, then removing effects of basic word form on word duration implicitly would indirectly eliminate effects of this optimization for the usual communicative scenario. If communication with a visible partner is the usual communicative scenario, then the control model would remove predictability effects that optimize for speech with a visible partner.

To understand why effects of Word Frequency disappeared, but effects of Preceding and Following Context persisted, remember from Chapter 2 that an information-theoretically optimal lexicon would select a phonological form for each word in proportion to the negative log of its probability, without adding any redundancy. By eliminating variation according to basic word form, with a linear model in log space, the control model essentially removes variation due to the entropy of such a unigram

source. Thus, the only variation in the residuals that can be systematically explained by Word Frequency is variation due to redundancy.

The control model does not, however, have any terms that would allow it to remove variation due to the entropy of a bigram, trigram, or higher-order source. Talkers may respect such dependencies in their static grammatical knowledge by, for example, maintaining a distinct representation of "got" in "got you."[6] If talkers do this, then our control model would be leaving in these influences of higher-order dependencies in the source, and the residuals would still contain variation according to these higher-order, non-redundancy based notions of predictability. Accordingly, while we see an effect of Preceding and Following Context in the Visible data, we don't necessarily see an effect of *redundancy* according to Preceding and Following Context.

Whether or not this account of which predictability effects weakened in the Visible speech data is correct, we have established that the differences between ADS and CDS cannot be attributed solely to visibility. Moreover, by showing that the amount of redundancy decreases in Visible speech, these results suggest that talkers adjust their redundancy strategy to take advantage of the increased channel capacity, suggesting a close relationship between predictability effects and communicative efficiency.

4.5 Conclusion

First, these results show that predictability effects exist in child-directed speech, so we are justified in trying to measure how useful they may be for language acquisition. Second, these results suggest that talkers modulate predictability effects according to at least very coarse, salient, and easy-to-track listener and channel characteristics. This second result more closely ties predictability effects to communication than previous studies, suggesting that an account of predictability effects that does not appeal to communicative efficiency, or at least the listener and channel elements of communication, is inadequate.

In the next two chapters, we will build probabilistic models that to exploit correlations between word duration and syntactic probabilities, and provide some evidence that the models succeed because of effects of syntactic probability on word duration.

[6] These distinct representations would be of the same kind that allow "got you" to be pronounced as [gatʃu], but forbid "got yellow" from being pronounced [gatʃɛlo^{w}].

Chapter 5

Acoustics for Chunking

5.1 Introduction

As a first computational step towards investigating how useful predictability effects could be for children trying to learn about syntax, this chapter develops statistical models for learning syntactic *chunks*: syntactic constituents with no internal structure. In addition to its simplicity, syntactic chunking is an interesting task because both Predictability Bootstrapping and Prosodic Bootstrapping strategies should, in principle, be reasonable. To compare these possibilities, we will first devise models that look for statistical relationships between syntactic chunks and either durational measures or prosodic annotations. When learning from words and prosodic annotations, the models will implement prosodic bootstrapping by design, and so provide a clear indication of what prosodic bootstrapping looks like in this task on this dataset. When learning from words and word durations, the models will implement "durational bootstrapping" by design, and we will assess the comparative feasibility of prosodic bootstrapping and predictability bootstrapping by comparing their behavior to the prosodic bootstrapping model.

In short, we find that learning from words and hand-annotated break index outperforms learning from words alone, and learning from words and word duration further outperforms learning from words and break index on one evaluation metric while matching learning from words and break index on the other. Moreover, break index is useful only when its relationship with syntactic structure is mediated by an intermediate variable, as expected in prosodic bootstrapping, but this intermediate variable is not important for learning from word duration. Together, these results will suggest that prosodic bootstrapping of syntactic chunks is feasible if the prosodic analysis is

already known, but that some other kind of statistical dependency, such as that underlying predictability effects, is useful when learning from word duration directly.

5.2 Syntactic Chunking

Syntactic chunking focuses on identifying non-overlapping syntactically-cohesive contiguous sequences of words, called chunks. In this chapter, we will be defining syntactic chunks to be just syntactic constituents with no internal constituents, but this definition conflicts somewhat with early approaches to syntactic chunking. Abney (1991) introduced the idea of syntactic chunks, defining a chunk to consist of a head word (usually a content word), any function words that the content word head selects, and a fully-connected syntactic analysis for these words. Abney cited two motivating intuitions for looking at the potential role of chunks in syntactic processing. First, there had been some recent laboratory work on so-called "performance structures" that appeared to integrate prosodic phrasing and syntactic constituency; we will discuss these shortly. Second, Abney noted that some parts of syntactic structure are fairly templatic and easy to describe with a context-free grammar, while other parts are more variable and lexically-specific. Under Abney's account, the templatic bits correspond to syntactic chunks, which are then combined according to the lexical properties of the chunk heads.

As an example, Figure 5.1 reproduces Figure (2) from Abney (1991), showing Abney's chunking analysis for "the bald man was sitting on his suitcase." This analysis builds up the subject NP, the core of the verb phrase, and the PP adjunct as chunks. In Abney's chunking parser, the output of the chunker is then handed off to an "attacher," which integrates the chunks into a tree by adding an arc from the IP node to the DP node, and from the lower VP node to the PP node. Under this approach, the chunker can rely on templatic, non-lexically-specific notions of DPs, PPs and VPs, while the attacher can use lexical features of the heads to determine that the PP is an adjunct rather than an argument.

In this example, two of the chunks are constituents, and one is not. Subsequent efforts to find and use syntactic chunks in a large-scale natural language processing context led to a number of refinements to the operational definition of "chunk" that were engineered for utility in NLP tasks rather than psychological plausibility. For example, Tjong et al. (2000) excluded all words to the right of a chunk's head from the chunk, and did not consider a potential chunk to be a chunk if it was contained within

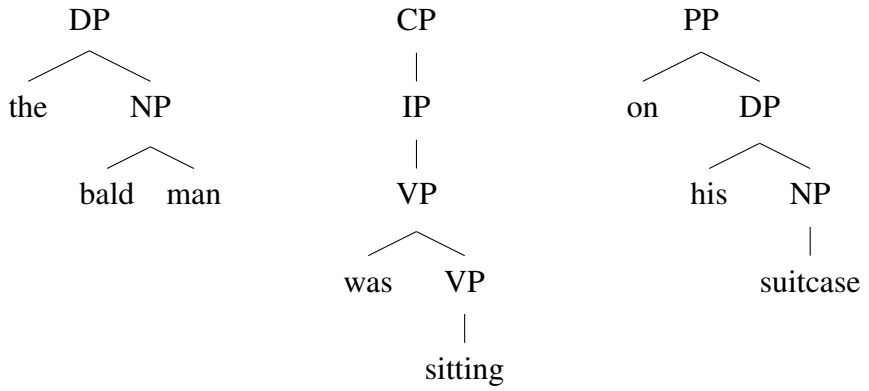


Figure 5.1: Chunking analysis for "the bald man was sitting on his suitcase," reproduced from Abney (1991).

a larger chunk. Some of these early NLP systems will be discussed in Section 5.3

As mentioned, Abney's other motivation for chunk-based syntactic processing lay in work on the "performance structures" of Gee and Grosjean (1983). A performance structure is a kind of tree structure built over words of a sentence based on how naïve subjects pause when reading the sentence. Specifically, to obtain a performance structure for a particular sentence, Gee and Grosjean told participants to read a sentence while pausing between each word. They then measured the duration of each pause as a percentage of the total time spent pausing, and used this measure (averaged over multiple repetitions of the same sentence at different speeds) as a measure of the strength of the word boundary. A tree structure can then be built by iteratively merging words separated by the shortest un-merged pause until no un-merged pauses remain.

Gee and Grosjean (1983) interpreted performance structures as a reflex of linguistic performance, as opposed to competence, and sought to show how performance structures could be predicted from syntactic structures. To do so, they presented a model of production[1] for building trees with pausing predictions that referred to both syntax and intermediate phrases.[2] Abney (1991) proposed that syntactic chunks *are* intermediate phrases, together with associated syntactic structure, and developed a set of rules for predicting boundary prominence from intermediate phrase-based syntactic chunks. Abney argued that the syntactic chunk-dependent notion of boundary prominence is a

[1] Gee and Grosjean claim that they present an "algorithm," not a model, but they seem to be worried about the difference between a computational-level model and an algorithmic-level model. Their model is at the computational level.

[2] They used the term "phonological phrase" from Selkirk's 1978 theory of prosody, but phonological phrases are basically the same as intermediate phrases in ToBI, which did not exist yet, and Gee and Grosjean claimed to not rely on the details of any one theory.

better fit to pausing data than the dominant one from the performance structure literature, which assumed a simpler view of syntactic complexity (essentially just counting nodes). Thus, Abney's "parsing by chunking" account leads naturally to a kind of Prosodic Bootstrapping account: prosodic cues help infants identify chunk boundaries, providing a basis for learning both chunk-internal structure and how chunks should be combined into a tree.

5.3 Models

5.3.1 Previous Work

Early work on syntactic chunking noted that chunking can be viewed as a tagging task: each word is either in or not in a chunk. In practice, most chunking systems have used the three-tag **IOB** tagset, signalling that a word is either **I**nside a chunk (but not at the beginning), **O**utside a chunk, or at the **B**eginning of a chunk. Exploiting this fact, many early approaches, all supervised, adapted the tagging formalisms developed for part-of-speech tags to produce chunking tags. Indeed, in the 2000 CoNLL shared task on chunking, Osborne (2000) adapted the supervised maximum entropy part-of-speech tagger of Ratnaparkhi (1996) as a syntactic chunker. Simply by providing the tagger with **IOB** tas as the supervised signal and adjusting the features, Osborne achieved an F-score of nearly 92% (the best systems in that evaluation achieved F-score of around 94%). Subsequent systems improved on the chunking-as-tagging approach, with Molina and Pla (2002) adapting Hidden Markov Models and Sha and Pereira (2003) adapting conditional random fields for this task.

Ponvert et al. (2010) and Ponvert et al. (2011) provide the only previous work towards unsupervised chunking. Ponvert et al. (2010) described a simple method for chunking that uses only bigram counts and punctuation; when the chunks are combined using a right-branching structure, the resulting trees achieve unlabeled bracketing precision and recall that is competitive with other unsupervised parsers. Ponvert et al. (2011), used unsupervised finite-state chunkers similar to ours both as chunkers and also in a cascade to provide a hierarchical parse. Their systems included special treatment of punctuation, and provided state-of-the-art results for unsupervised lexicalized parsing on some newswire datasets.

5.3.2 Our models

We will also formulate chunking as a tagging task. We use Hidden Markov Models (HMMs) and their variants to perform the tagging, with carefully specified tags and constrained transition distributions to allow us to interpret the results as a bracketing of the input. Specifically, we use four chunk tags: **B** ("Begin") and **E** ("End") tags are interpreted as the first and last words of a chunk, respectively, with **I** ("Inside") corresponding to other words inside a chunk and **O** ("Outside") to all other words. The transition matrices are constrained to afford 0 probability to transitions that violate these definitions. Additionally, the initial probabilities are constrained to forbid the models from starting inside or at the end of a phrase.

We use this four-tag **OBIE** tagset rather than the three-tag **IOB** tagset for two reasons. First, the **OBIE** set forces all chunks to be at least two words long (the shortest chunk allowed is **B E**). Imposing this requirement allows us to characterize the task in concrete terms as "learning when to group words together." Second, as we seek to incorporate acoustic correlates of prosody into chunking, we expect edge behavior to merit explicit modeling.[3]

(a)	Words	g.aa	dh.ae.t.s	dh.ae.t	s.aw.n.d.z	p.r.ih.t.iy	b.ae.d	t.ax	m.iy
	Acoustics	4	4	6	4	5	4	5	6
	ToBI	1	2	1	1	1	1	1	3
(b)		O	O	B	I	I	E	B	E
(c)				(			)	(	)
(d)	(			(			)	(	))

Figure 5.2: (a) Example input sequences for the sentence "Go[d] that's that sounds pretty bad to me," demonstrating the three types of input (phonetic word transcriptions, acoustic clusters, and ToBI break indices). (b) Example output tags. (c) The bracketing corresponding to (b). (d) The flat tree built from (b).

In the following subsections, we describe the various models we use. Note that input to all models is discrete, consisting of words, ToBI annotations, and/or discretized acoustic measures (we describe these measures and their discretization in Section 5.3.5). See Figure 5.2 for examples of system input and output; different models will receive different combinations of the three kinds of input.

[3]Indeed, when we tried using the **IOB** tag set in preliminary experiments, dev-set performance dropped substantially, supporting this latter intuition.

5.3.3 Baseline Models

Our baseline models are all standard HMMs, with the graphical structure shown in Figure 5.3(a). The first baseline uses *lexical* information only; the observation at each time step is the phonetic transcription of the current word in the sentence. To handle unseen words at test time, we use an "UNK." token to replace all words in the training and evaluation sets that appear less than twice in the training data. Our second baseline uses *prosodic* information only; the observation at each time step is the hand-annotated ToBI Break Index for the current word, which takes on one of seven values: { 0, 1, 2, 3, 4, X, None }.[4] Our final baseline uses *acoustic* information only. The observations are one of six automatically determined clusters in an acoustic space, as described in Section 5.3.5.

All models were trained to a Maximum Likelihood Estimate (MLE). Conceptually, MLE seeks to find the single set of parameter estimates $\hat{\theta}$ that best explains the observed data D:

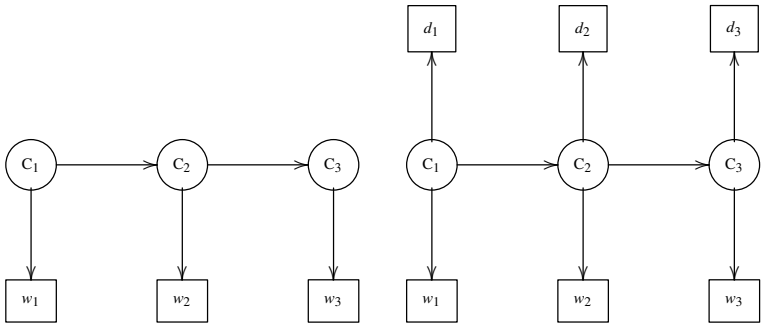
$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, P(D|\theta)$$

Our observed data for the baseline models will be individual sequences of words, word durations, or annotated break index. For the combined models (detailed in the next section) it will be pairs of such sequences.

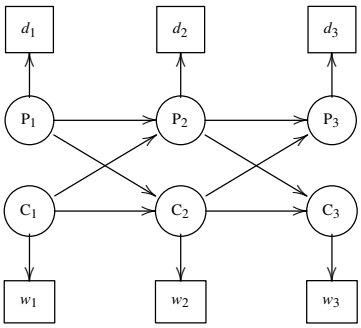
It is intractable to compute the maximum exactly, so we used Expectation-Maximization (EM). EM is an iterative procedure that adjusts the parameter set in each iteration in the direction of the gradient of the probability manifold with respect to model parameters. EM training usually proceeds until the probability changes little from one iteration to the next, which indicates that we have reached a place in the parameter space where the gradient of the probability manifold with respect to the parameters is close to zero. Such a location is hopefully a large local maximum (although it could be a minor local maximum, or even a saddlepoint).

EM is called EM because it involves an Expectation (E) and a Maximization (M) step. In an Expectation step at iteration n, we hold model parameters θ^n fixed and compute the probability of each hidden and observed variable in our training set. In a Maximization step, we hold those expected counts fixed and treat them as observations in estimating new parameters.

[4] The numerical break indices indicate breaks of increasing strength, "X" represents a break of uncertain strength, and "None" indicates that the preceding word is outside one of the fluent prosodic phrases selected for annotation. Additional distinctions marked by "-" and "p" were ignored.



(a) Standard HMM (HMM) (b) Two Output HMM (THMM)



(c) Coupled HMM (CHMM)

Figure 5.3: Graphical structures for our various HMMs. c_i nodes are constrained using the **OBIE** system, p_i nodes are not. w_i nodes represent lexical outputs, and d_i nodes represent acoustic or ToBI outputs. (Rectangular nodes are observed, circular nodes are hidden).

We trained our baseline HMMs using the Baum-Welch algorithm, which is just EM that uses the Forward Backward algorithm in the E-step and estimates the parameters of a Plain HMM in the M-step. We used the Viterbi algorithm for inference.[5]

5.3.4 Combined Models

As discussed in Section 3.2.3.2, previous theoretical and experimental work suggests a combined model which models uncertainty both between prosody and acoustics, and between prosody and syntax. To measure the importance of modeling these kinds of uncertainty, we will evaluate a series of model structures that gradually divorce acoustic-prosodic cues from lexical-syntactic cues.

Our first model is the standard HMM from Figure 5.3(a), but generates a (word,

[5]We actually used the junction tree algorithm from the GRMM portion of MALLET(McCallum, 2002; Sutton, 2006), which, in the special case of an HMM, reduces to the Forward-Backward algorithm when using Sum-Product messages, and to the Viterbi algorithm when using Max-Product messages.

acoustics) or (word, ToBI) pair at each time step. This model has the simplest structure, but includes a separate parameter for every unique (state, word, acoustics) triple, so may be too unconstrained to learn anything useful.

To reduce the number of parameters, we propose a second model that assumes independence between the acoustic and lexical observations, given the syntactic state. We call this a "Two-output HMM (THMM)" and present its graphical structure in Figure 5.3(b). It is straightforward to extend Baum-Welch to accommodate the extra outputs of the THMM.

Finally, we consider a model that explicitly represents prosodic structure distinctly from syntactic structure with a second sequence of tags. We use a Coupled HMM (CHMM) (Nefian et al., 2002), which models a set of observation sequences using a set of hidden variable sequences. Figure 5.3(c) presents a two-stream Coupled HMM for three time steps. The model consists of an initial state probability distribution π_s for each stream s, a transition matrix a_s for each stream s conditioning the distribution of stream s at time $t+1$ on the state of both streams at time t, and an emission matrix b_s for each stream conditioning the observation of stream s at time t on the hidden state of stream s at time t.[6]

Intuitively, the states emitting acoustic measures operationalize prosodic structure, and the states emitting words operationalize syntactic structure. Crucially, Coupled HMMs impose no *a priori* correspondence between variables of different streams, allowing our "syntactic" states to vary freely from our "prosodic" states. As two-stream CHMMs maintain two emission matrices, two transition matrices, and two initial state distributions, they are more complex than the other combined models, but more closely embody intuitions inspired by previous work on the prosody-syntax interface.

Like the Baseline HMMs, our Coupled HMMs were trained using EM. In the E-step, the marginals were computed using the implementation of the junction tree algorithm available in MALLET (McCallum, 2002; Sutton, 2006). The junction tree algorithm is just a more general version of the Forward-Backward algorithm mentioned above for Plain HMMs. In the M-step, model parameters were updated according to the following equations.

For each stream s, initial state probabilities are updated according to:

$$\pi_s(q=h) = \frac{\sum_{u \in C} P_u(q_0^s = h)}{\sum_{u \in C} \sum_{i \in H_s} P_u(q_0^s = i)}$$

[6] We explored a number of minor variations on this graphical structure, but preliminary experiments yielded no improvement.

where H_s is the set of hidden states for stream s, C is the corpus of utterances u, q_t^s is the variable of stream s at time t, and P_u is the table of marginals for utterance u. Similarly, for a two-stream CHMM with streams A and B, transitional probabilities are updated according to:

$$a_s(q = h|q^A = i, q^B = j) = \frac{\sum_{u \in C} \sum_{t=1}^{T} P_u(q_t^s = h, q_{t-1}^A = i, q_{t-1}^B = j)}{\sum_{u \in C} \sum_{t=1}^{T} \sum_{k \in H_s} P_u(q_t^s = k, q_{t-1}^A = i, q_{t-1}^B = j)}$$

where T is the number of words in the utterance. The update for emission probabilities is the same as in a standard HMM.

During test, the Viterbi tag sequence for each model is obtained by simply replacing the sum-product messages with max-product messages.

5.3.5 Acoustic Cues

As explained in Section 3.2.3.2, prosody is an abstract hidden structure which only correlates with observable features of the acoustic signal, and we seek to select features which are both easy to measure and likely to correlate strongly with the hidden prosodic phrasal structure. While there are many possible cues, we have chosen to use duration cues. These should provide good evidence about prosodic phrases due to the phenomenon of pre-boundary lengthening (e.g. Beckman and Edwards, 1990; Wightman et al., 1992), wherein words, and their final rime, lengthen phrase-finally. This is likely especially useful for English due to the lack of confounding segmental duration contrasts (although variation in duration is unpredictably distributed (Klatt, 1976)), but should be useful in varying degrees for other languages.

We gather five duration measures:

1. Log total word duration: The annotated word end time minus the annotated word start time.

2. Log onset duration: The duration from the beginning of the word to the end of the first vowel.

3. Log offset duration: The duration from the beginning of the last vowel to the end of the word.

4. Onset proportion consonant: The duration of the non-vocalic portion of the word onset divided by the total onset duration.

5. Offset proportion consonant: The duration of the non-vocalic portion of the word offset divided by the total offset duration.

If a word contains no canonical vowels, then the first and last sonorants are counted as vocalic. If a word contains no vowels or sonorants, then the onset and offset are the entire word and the proportion consonant for both onset and offset is 1 (this occurred for 186 words in our corpus).

The potential utility of this acoustic space was verified by visual inspection of the first few PCA components, which are the straight lines through the high-dimensional acoustic space that represent the most acoustic variation. These components suggested that the position of a word in this acoustic space correlated with bracket count. The plot in Figure 5.4 presents $3,000$ randomly selected words in the first two components of the exploratory PCA, and we see that words with more following brackets tend to be on the left side of the PCA analysis, and subsequent components (not presented) appear to increase separability. We discretize the raw (i.e. non-PCA) space with k-means with six initially random centers.

5.4 Experiments

5.4.1 Dataset

All experiments were performed on part of the Nite XML Toolkit edition of the Switchboard corpus (Calhoun et al., 2010). Specifically, we gathered all conversations which have been annotated for syntax, ToBI, and Mississippi State phonetic alignments (which lack punctuation).[7] The syntactic parses, word sequences, and ToBI break indices were hand-annotated by trained linguists, while the Mississippi State phonetic alignments were automatically produced by a forced alignment of the speech signal to a pronunciation-dictionary based phone sequence, providing an estimate of the beginning and end time of each phone. A small number of annotation errors (in which the beginning and end times of some phones had been swapped) were corrected by hand. This corpus has 74 conversations with two sides each.

[7] We threw out a small number of sentences with annotations errors, e.g. pointing to missing words.

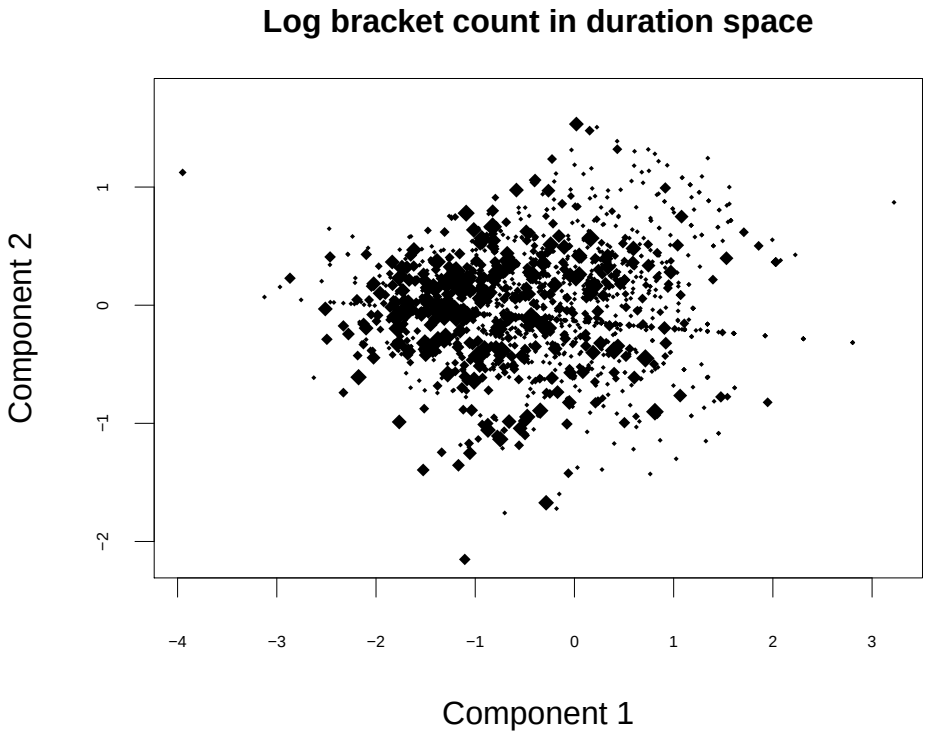


Figure 5.4: Words in the first two principle components of our acoustic space. Each point is scaled in size according to the log number of closing brackets following the word.

We split this corpus into an 80%/10%/10% train/dev/test [8] partition by dividing the entire corpus into ten-sentence chunks, assigning the first eight to the training partition, and the ninth and tenth to the dev and test partitions, respectively. We then removed all sentences containing only one or two words. Sentences this short have a trivial parse, and are usually formulaic discourse responses (Bell et al., 2009), which may influence their prosody. The final corpus statistics are presented in Table 5.1.

[8]The dev set was used to explore different model structures in preliminary experiments; all reported results are on the test set.

	Train	Dev	Test
Words	68,533	7,981	8,746
Sentences	6,420	778	802

Table 5.1: Data set statistics

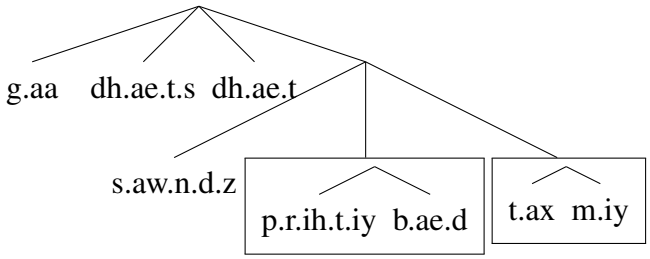


Figure 5.5: Example gold-standard with clumps in boxes.

5.4.2 Evaluation

We use the Penn Treebank parsed version of Switchboard for evaluation. This version uses a slightly different tokenization from the Mississippi State transcriptions that were used as input to the models, so we transformed the Penn treebank tokenization to agree with the Mississippi State tokenization. This transformation primarily involved concatenating clitics to their base words—i.e. "do" and "'nt" into "don't"—and splitting multi-word expressions. We also removed all gold-standard nodes spanning only `Trace` or `PUNC` (recall that the input to the models did not include punctuation) and collapsed all unary productions.[9] In all evaluations, we convert our models' output tag sequence to a set of matched brackets by inserting a left bracket preceding each word tagged **B** tag and a right bracket following each word tagged **E**. This procedure occasionally results in a sentence with an unmatched opening bracket. If the unmatched opening bracket is one word from the end of the sentence, we delete it, otherwise we insert a closing bracket at the end of the sentence. Figure 5.2 shows example input sequences together with example output tags and their corresponding bracketings.

Previous work on chunking, most notably the 2000 CONLL shared task (Tjong et al., 2000), has defined gold standard chunks that are useful for finding grammatical relations but which do not correspond to any particular linguistic notion. It is not clear that such chunks should play a role in language acquisition, so instead we evaluate against traditional syntactic constituents from Penn Treebank-style parses in two different ways.

Our first evaluation method compares the output of the chunkers to what Ponvert et al. (2010) call *clumps*, which are just syntactic constituents that span only terminals. We created our clump gold-standard by taking the parse trees resulting from the preprocessing described above and deleting nodes that span a non-terminal. Figure 5.5

[9] As we evaluate unlabeled bracketing precision and recall, the label of the resulting nodes is irrelevant.

presents an example gold-standard parse tree with the clumps in boxes. This evaluation avoids penalizing chunkers for not positing hierarchical structure, but rewards chunkers only for finding very low-level structure.

In the interest of making no *a priori* assumptions about the kinds of phrases our unsupervised method recovers, we also evaluate our completely flat, non-recursive chunks directly against the fully recursive parses in the treebank. To do so, we turn our chunked utterance into a flat tree by simply putting brackets around the entire utterance as in Figure 5.2(d). This evaluation penalizes chunkers for never positing hierarchical structure, but makes no assumptions about which kinds of phrases ought to be found.

5.4.3 Models and training

In all, nine HMM models, two versions of the Common Cover Link (CCL) parser, and a uniform right-branching baseline were evaluated. The CCL Parser of Seginer (2007) is an unsupervised lexical model for fully-hierarchical constituency parsing. It has achieved state-of-the-art constituency parsing results on standard datasets in a variety of languages, such as the Wall Street Journal portion of the Penn Treebank, the Chinese Treebank, and the German Negra Treebank. It is included here as a state-of-the-art benchmark.

Three of our own HMM models were standard HMMs with chunking constraints on the four hidden states (as described in Section 5.3.4) that received as input either words, ToBI break indices, or word duration cluster information. These were intended as baselines to illuminate the utility of each information source in isolation. We also ran two each of Coupled HMM and Two-output HMM models that received words in one observed chain and either ToBI break index or duration cluster in the other observed chain. In the CHMM models, chunking constraints were enforced on the chain generating the words, while variables generating the duration or ToBI information ranged over four discrete states with no constraints.[10] All non-zero parameters were initialized approximately uniformly at random,[11] and we ran EM until the log corpus probability changed less than 0.001%, typically for 50-150 iterations.

The CCL parser was trained on the same word sequences provided to our models. We also evaluated the CCL parser as a clumper (CCL Clumper) by removing internal

[10] We also tried imposing chunking constraints on the second chain, but dev-set performance dropped slightly.

[11] In preliminary dev-set experiments, different random initializations performed within two points of each other.

Condition			Prec	Rec	F-sc
Baselines	HMM	Wds	23.5	39.9	26.3
		BI	7.2	4.8	5.8
		Ac	4.7	2.5	3.3
Combined Models	HMM	Wds+BI	24.4	22.2	23.2
		Wds+Ac	20.7	22.7	21.7
	THMM	Wds+BI	18.2	19.6	18.9
		Wds+Ac	36.1	47.8	**41.2**
	CHMM	Wds+BI	25.5	36.3	29.9
		Wds+Ac	33.6	**48.1**	39.5
CCL		Parser	15.4	41.5	22.4
		Clumper	**36.8**	37.9	37.3

Table 5.2: Scores for all models, evaluated on clumps. Input is words (Wds), break indices (BI), and/or acoustics.

nodes spanning a non-terminal. The right-branching baseline was generated by inserting one opening bracket in front of all but the last word, and closing all brackets at the end of the sentence.

5.4.4 Results

Table 5.2 presents results for our flat chunkers evaluated against Ponvert et al. (2010)-style clumps. Several points are apparent. First, all three HMM baselines yield very poor results, especially the prosodic baselines, whose precision and recall are both below 10%. Although the best combined models still have relatively low performance, it is markedly higher than either of the individual baselines, and also higher than the clumps identified by the CCL parser. Particularly notable is the fact that lexical and prosodic information appear to be super-additive in some cases, yielding combined performance that is higher than the sum of the individual scores. Not all combined models work equally well, however: the poor performance of the HMM combined model supports our initial hypothesis that it is over parameterized. Interestingly, our acoustic clusters work better than break indices when combined with words. Moreover, we do observe a difference between the THMM and the CHMM when learning from words and Break Index: the CHMM clearly outperforms the baseline models, but the THMM does not. This indicates that decoupling prosodic structure from syntactic

			% Covered		$\frac{words}{chunk}$	$\frac{chunk}{utt}$
Condition			Words	Utts		
Baselines	HMM	Wds	81.9	98.4	3.16	2.82
		BI	68.2	68.1	4.95	1.50
		Ac	46.3	71.1	4.18	1.21
Combined Models	HMM	Wds+BI	79.8	98.3	4.30	2.02
		Wds+Ac	83.3	98.5	3.71	2.45
	THMM	Wds+BI	84.6	99.0	3.84	2.40
		Wds+Ac	68.0	96.1	2.52	2.94
	CHMM	Wds+BI	83.1	99.0	2.86	3.17
		Wds+Ac	76.5	97.6	2.62	3.19
CCL		Clumper	48.3	99.9	2.30	2.29

Table 5.3: % words in a chunk, % utterances with > 0 chunks, and mean chunk length and chunks per utterance.

structure is important. Finally, we see that the THMM and CHMM obtain similar performance using words and acoustics, suggesting that modeling word duration patterns separately from syntactic structure is not important.

To provide further intuition into the kinds of chunks recovered by the different models, we list some relevant statistics in Table 5.3. These statistics show that the models using lexical information identify at least one chunk in virtually all utterances, with the better models averaging 2-3 chunks per utterance of around 3 words each. In contrast, the unlexicalized models find longer chunks (4-5 words each) but far fewer of them, with about 30% of utterances containing none at all.

We turn now to the models' performance on full parse trees, shown in Table 5.4. Two different scores are given for each system: the first includes the top-level bracketing of the full sentence (which is standard in computing bracketing accuracy, but is a free true positive), while the second does not (for a more accurate picture of the system's performance on ambiguous brackets). Comparing the second set of scores to the clumping evaluation, recall is much lower for all the chunkers; the relatively small increase in precision indicates that the chunkers are most effective at finding low-level structure. For both sets of scores, the relative F-scores of the chunkers are similar to the clumping evaluation, with the words + acoustics versions of the THMM and CHMM scoring best. Not surprisingly, the CCL parser has much higher recall

Condition			Prec	Rec	F-sc
Baselines	HMM	Wds	48.8(32)	26.3(15)	34.2(20)
		BI	52.4(21)	18.5(5)	27.3(8)
		Ac	52.5(15)	16.3(3)	24.9(5)
Combined Models	HMM	Wds+BI	54.4(32)	23.2(11)	32.5(16)
		Wds+Ac	51.0(32)	24.7(13)	33.3(18)
	THMM	Wds+BI	55.9(38)	26.8(15)	36.2(21)
		Wds+Ac	55.8(41)	31.0(20)	39.9(27)
	CHMM	Wds+BI	48.4(32)	28.4(17)	35.8(22)
		Wds+Ac	54.1(40)	31.9(21)	**40.1**(**28**)
CCL		Parser	38.2(28)	**37.6**(**28**)	37.9(**28**)
		Clumper	**58.8**(**42**)	27.3(16)	37.3(23)
Right-Branching			42.2(36)	64.8(59)	51.1(45)

Table 5.4: Model performance, evaluated on full trees. Scores in parentheses were computed after removing the full sentence bracket, which provides a free true positive.

than the chunkers, though the best chunkers have much higher precision. The result is that, using standard Parseval scoring (first column), the best chunkers outperform CCL on F-score; even discounting the free sentence-level bracket (second column) they do about as well.

It is worth noting that, although CCL achieves state-of-the-art performance on the English WSJ and German Negra corpora (Seginer (2007) reports 75.9% F-score on WSJ10, for example), its performance on our corpus is far lower and does not outperform a uniform right-branching baseline. This suggests that our corpus is significantly more difficult than WSJ, probably due to disfluencies and/or lack of punctuation.[12] We will see in Chapter 6 that the CCL exhibits degraded performance on a corpus of child-directed speech which is free from disfluencies but also lacks punctuation. This indicates that the lack of punctuation in this dataset is a significant source of error for the CCL parser. There are too many differences between the CDS dataset and this one to attribute all of the degradation to lack of punctuation, however. Moreover, we stress that the use of a right-branching baseline, while useful as a measure of overall performance, is not plausible as a model of language acquisition since it is highly

[12] Including punctuation improves CCL little, possibly because the punctuation in this corpus is nearly all sentence-final.

language-specific.

5.5 Discussion

Taken together, our results indicate that a purely local model that combines lexical and acoustic-prosodic information in an appropriate way can identify syntactic phrases far more effectively than a similar model using either source of information alone. Our best combined models outperformed the baseline individual models by a wide margin when evaluated against the lowest level of syntactic structure, and their performance was comparable to CCL, a state-of-the-art unsupervised lexicalized parser, when evaluated against full parse trees. It is disappointing that all of these systems scored worse than a right-branching baseline, but this result underscores the major differences between parsing spoken utterances (even using transcriptions) and parsing written text (where CCL and other unsupervised parsers were developed and tested). Since children learning language do not (at least initially) know the head direction of their language, the right-branching baseline for English is not available to them. Thus, combining lexical and acoustic cues may provide them with initial useful information about the location of syntactic phrases, as suggested by the prosodic bootstrapping and predictability bootstrapping hypotheses.

Intriguingly, the evaluation on Ponvert et al.-style clumps gave some indication that the HMMs were capable of exploiting both prosodic and predictability based cues. When we gave the combined models words and break index, the CHMM outperformed the baseline models, while the THMM did not. This result indicates that syntax and prosody co-vary in a way that demands in intermediate representation, as predicted by prosodic bootstrapping (while the models may have used this representation to differentiate between prosodic boundaries that were and were not likely to be syntactic boundaries, as hypothesized in Section 3.2.3, we did not constrain them to do so). However, when we gave the combined models word duration information there was no clear difference between the THMM and CHMM in performance. Since predictability bootstrapping does not involve an intermediate structure, this suggests that the models were doing something like predictability bootstrapping when given word duration.

This pattern of results suggests that the strongest source of regularities in word duration is predictability effects, not prosodic phrasing (at least when inducing chunk tags). There are two, mutually-compatible, possible explanations for this. The first explanation is inspired by the finding by Seidl (2007), discussed in Section 3.2.3.3,

that 6-months-olds identify clausal boundaries on the basis of both word duration and intonation, but not word duration alone. This explanation proposes that it is impossible to identify prosodic phrase breaks reliably on the basis of word duration alone, and so the strongest source of regularity in word duration alone is predictability effects. Under this explanation, it is still possible that the strongest source of regularities in syntactic chunks, word duration and fundamental frequency variation is prosodic structure. If so, we would still expect a CHMM that learned from both word duration and fundamental frequency traces to outperform a THMM that learned from word duration alone.

The second possible explanation is inspired by the observation that the models that learned from words and word duration substantially outperformed the models that learned from words and break index on the Ponvert et al.-style clumps evaluation. This explanation proposes that chunking models engage in predictability bootstrapping, rather than prosodic bootstrapping, because predictability effects provide more information about syntactic structure than prosodic phrasing does.

If the second explanation is correct, then our models have actually overestimated the utility of prosodic information. Section 5.4.2 mentioned that we converted the Penn treebank tokenization to the Mississippi State tokenization. Because this transformation involved concatenating clitics to their base words, it eliminated many word boundaries that were not prosodic phrase boundaries but were syntactic phrase boundaries. For example, "I'm" is a frequent word, and contains a boundary that will virtually always be annotated 0 under American English ToBI but also contain a syntactic boundary between the subject NP and VP. It is difficult to assess *how much* the models overestimated the utility of prosody, however. This kind of regularity is systematic, and so might be easy to learn.

On balance, these results favor the second explanation. The first explanation proposes that the word and word duration models engage in predictability bootstrapping because prosodic phrasing is underspecified in the input, since we're missing intonation. However, models which learn from words and break index observe a direct encoding of the prosodic phrasing. The words and break index models, then, observe exactly those aspects of intonation that are relevant to prosodic phrasing, and serve as an upper-bound for an unsupervised HMM chunker that engages in prosodic bootstrapping from words, word duration, and intonation. Because this upper-bound is outperformed by the models that learn from only words and word duration, we can conclude that the words and word duration models were not simply relying on a noisier encoding of prosodic structure.

In summary, these results show that acoustic cues can be useful for identifying syntactic structure when combined with lexical information. Moreover, they provide good reason to think that predictability bootstrapping is at least as useful for syntax as prosodic bootstrapping, and motivate further investigation of predictability bootstrapping.

However, these results alone have several shortcomings. All models achieved fairly low performance, and were limited to proposing flat structures only. Nested hierarchy and recursion play a central in syntactic theory, and ideally, a model of syntax acquisition should be powerful enough to model nested structures and achieve high performance. Additionally, all models were evaluated on adult-directed speech (ADS), which is different from child-directed speech (CDS) in many ways. Most notably, CDS is much less disfluent than ADS. Chapter 6 will address all of these issues by performing experiments with unsupervised dependency parsing on CDS.

Chapter 6

Acoustics for Dependencies

6.1 Introduction

The previous chapter presented a preliminary attempt to use predictability effects (and prosodic phrase structure) in a constrained, unsupervised grammar induction task. Experiments showed that words and word duration were substantially more helpful than just words alone or words and break index, and the resulting systems outperformed a state-of-the-art lexicalized parser (although they did not outperform a language-specific right-branching baseline). This chapter presents a similar study, looking to learn a fully-hierarchical parser from words and word durations rather than only a chunker.

This chapter focuses on exploiting predictability effects as a direct cue to syntactic structure. It does so by incorporating word duration information into models for unsupervised dependency parsing. The strategy, then, is similar to the previous chapter: design a system that implements durational bootstrapping, measure the utility of word duration measures, and assess the extent to which the statistical dependency between word duration and syntax reflects predictability effects and/or prosodic structure. The primary contribution of this chapter is evidence that even a very simple representation of word duration is useful for unsupervised dependency parsing. This chapter also presents the first attempt at unsupervised dependency parsing of transcribed speech rather than text, along with new, hand-annotated development and test sets for child-directed speech.

The models used in this chapter are based on the Dependency Model with Valence (DMV) of Klein and Manning (2004), first discussed in Chapter 3. Since children learn from speech, this will involve running the DMV on words and word durations

from speech, while it was originally developed on Part Of Speech (POS) tags from newspaper text. Since there are many more words than there are POS tags, our models will be much more sparse. Headden et al. (2009) presents a Bayesian variant of the DMV, which will form the basis of our models, that uses the smoothing inherent in Bayesian approaches along with backoff techniques to mitigate data sparsity.

The rest of this chapter is organized as follows. Section 6.2.2 describes the Bayesian DMV with Backoff presented by Headden et al. (2009), and Section 6.2.3 describes how the Bayesian DMV with Backoff is modified to learn from words and word durations. Section 6.3 evaluates the models on three types of language (described in Section 6.3.1).

6.2 Models

This section describes our parsing models as elaborations on the Dependency Model with Valence (DMV, Klein and Manning, 2004). Chapter 3, in particular Section 3.3.2, provides a non-technical, intuitive introduction to the DMV. This section will describe the DMV in greater technical detail, an elaboration of the DMV to include Backoff will be described in Section 6.2.2, and our elaboration on the DMV with Backoff in Section 6.2.3.

6.2.1 The Dependency Model with Valence

The DMV is a generative model for dependency trees in which each dependent is generated by its head, and the root of the dependency tree is drawn from a P_{root} probability distribution. Specifically, we first select the root word of the sentence w with probability $P_{root}(w)$. Next, when considering a head h, we look in a direction dir (either to the left or to the right), and consider whether to generate another Dependent or a Boundary. We generate a Dependent with probability $P_{gen}(\text{Dep}|h, dir, val)$, and a Boundary with probability $P_{gen}(\text{Bound}|h, dir, val)$. val encodes the valence of h; if this is the first dependent for h in the direction of dir, val is F, otherwise it is T. If we decide to generate a Boundary, we consider no more dependents in the direction of dir, and we say that h is *sealed* in that direction. If we decide to generate a Dependent, we decide which Dependent d we should choose with probability $P_{choose}(d|h, dir)$. The gen decision is assumed to be independent of the choose decision, so the probability of a head h generating a dependent d is just the product $P_{gen}(\text{Dep}|h, dir, val)P_{choose}(d|h, dir)$.

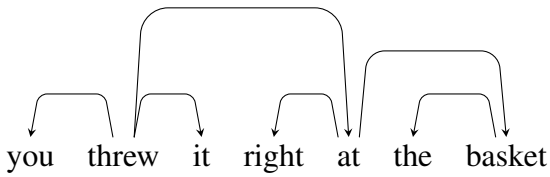


Figure 6.1: Example unlabeled dependency parse, reproduced from Figure 3.6.

If we restrict ourselves to projective dependency trees, the DMV can be expressed as a Probabilistic Context Free Grammar by decorating nodes with the valence of the head and the current direction of attachment. For example, the rule $\overrightarrow{h_F} \rightarrow \overrightarrow{h_T} \quad \overline{d}$ encodes a head h generating d (which is stopped in both directions, indicated by the straight line above) as its first dependent (since the subscript of the head child is T) to the right (the direction of the arrows). The full probability of this dependency is the product of not stopping and choosing d as the dependent: $P(\text{Dep}|h,\rightarrow,T)P(d|h,\rightarrow)$. Notice that the parent of this rule is $\overrightarrow{h_F}$, with a subscripted valence of F. This valence encodes the fact that h now has a dependent to the right and subsequent rightward P_{gen} decisions should be conditioned on $val = F$.

The decision to generate a Boundary, on the other hand, is represented by a unary rule. Since dependents to the right are independent of dependents to the left (given the head word), the order in which we generate dependents on each side does not matter. The probability of a tree will be the same whether we generate to the right first, to the left first, alternate between generating to the right and left, or even decide randomly whether to look right or left after each decision (as long as we decorate our nodes appropriately with the valence in each direction and to ensure that we stop looking in a particular direction once we've generated a Boundary in that direction).[1]

For convenience, we will follow Klein and Manning (2004) and always generate to the left first. The first Boundary decision switches from generating dependents to the right to generating dependents to the left: $\overleftarrow{h_F} \rightarrow \overrightarrow{h_{val}}$, at probability $P(\text{Bound}|h,\rightarrow,val)$. The second Boundary decision fully seals the head, allowing it to be generated as the dependent of some other word: $\overline{h} \rightarrow \overleftarrow{h_{val}}$ at probability $P(\text{Bound}|h,\leftarrow,val)$. Figure 6.2 shows the full DMV representation of the dependencies of the last three words of Figure 3.6, reproduced in Figure 6.1.

[1] Indeed, the irrelevance of attachment order is emphasised by the more efficient *split-head* PCFG formulation used by Headden et al. (2009), which splits each putative head into two non-terminals, one of which takes dependents to the left and one takes dependents to the right.

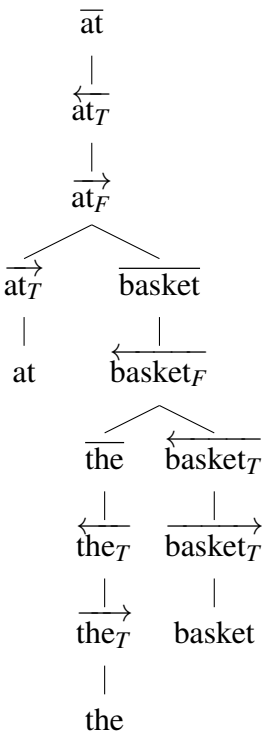


Figure 6.2: The last three words of Figure 3.6 in the Context Free Grammar used by the DMV.

Consider the sentence "I jumped." Let's step through the generative process for this sentence, following the dependency tree in Figure 6.3.

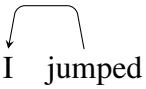
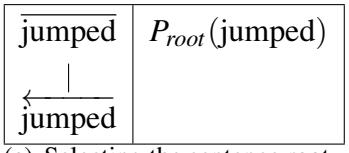


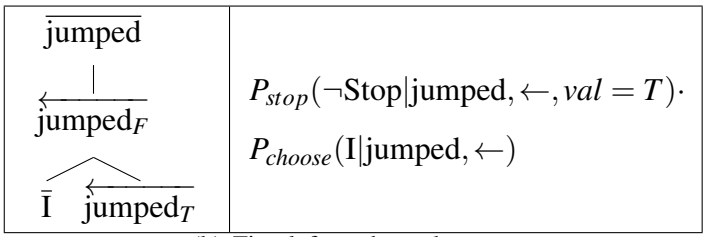
Figure 6.3: Dependency tree for "I jumped."

First, we generate the root "jumped" of the sentence with probability $P_{root}(\text{jumped})$, reflecting the likelihood of "jumped" to be a sentence root. In a good model, this probability will be high, since "jumped" is a verb and so readily serves as the root of a sentence. This leads to the subtree in Figure 6.4(a).

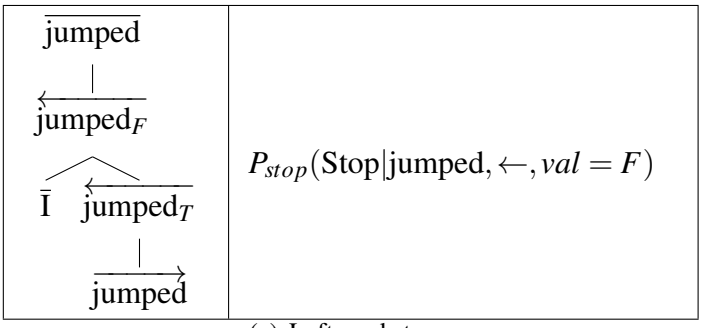
Next, we look to generate dependents to the left. We decide to generate a Dependent with probability $P_{gen}(\text{ Dep} \mid \text{jumped}, \leftarrow, val = T)$, reflecting the probability that "jumped" takes a dependent to the left, given that this would be the first dependent to the left. In a good model, this will again be high, since, bizarre text from computational linguistics dissertations excepted, "jumped" almost always takes a noun phrase subject



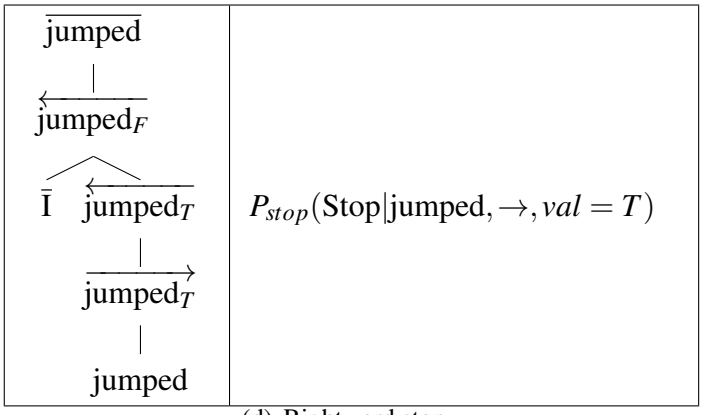
(a) Selecting the sentence root.



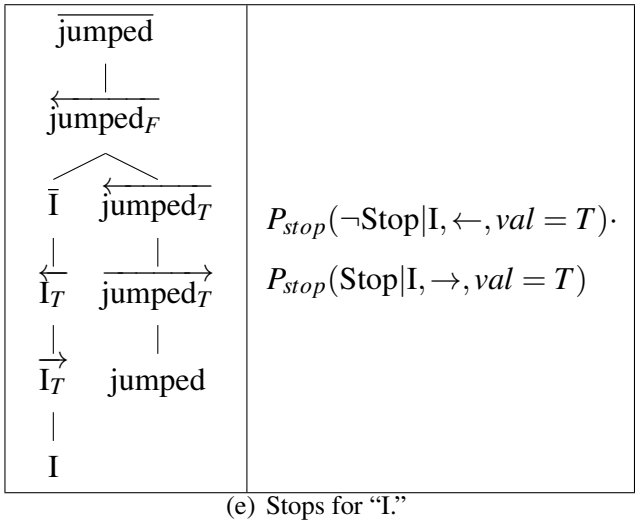
(b) First leftward attachment.



(c) Leftward stop.



(d) Rightward stop.



(e) Stops for "I."

to its left. The word "I" is chosen as the dependent at probability $P_{choose}(\text{I}|\text{jumped}, \leftarrow)$. This probability should also be high, as "I" is an appropriate leftward dependent for

"jumped." This produces the subtree in Figure 6.4(b)

We then generate a Boundary to the left, reflecting the likelihood that "jumped" does not take a dependent to the left given that the hypothetical dependent would not be the first leftward dependent. This leads to the subtree in Figure 6.4(c) and happens with probability $P_{gen}(\text{ Dep} \mid \text{jumped}, \leftarrow, val = F)$

Now that "jumped" has a leftward Boundary, we look rightward, and generate a rightward Boundary with probability $P_{gen}(\text{Bound} \mid \text{jumped}, \rightarrow, val = T)$. As "jumped" is now sealed in both directions, we also write down the word without any direction or valence annotations, leading to the subtree in Figure 6.4(d).

Finally, we generate leftward and rightward Boundaries for "I" with no dependents with probabilities $P_{gen}(\text{Bound} \mid \text{I}, \leftarrow, val = T)$ and $P_{gen}(\text{Bound} \mid \text{I}, \rightarrow, val = T)$, producing the final tree in Figure 6.4(e). The probability of the full tree is just the product of the factors introduced on the right side of each table.

The original DMV, as presented in Klein and Manning (2004), was trained using Expectation-Maximization, and learned from part of speech tags rather than words. Specifically, expected counts were computed using the Inside-Outside algorithm, and, for dependent d, head h, lexicon $\mathcal{L}$ and triple (d,h,dir) representing an arc from h to d in the direction of dir, the update equation for the estimated choose distribution $\hat{P}_{choose}$ after iteration n was:

$$\hat{P}^{n+1}_{choose}(d|h,dir) = \frac{E^n(d,h,dir)}{\sum_{c\in\mathcal{L}}(E^n(c,h,dir))}$$

Similarly, for gen decision g, valence v, and quadruple (s,h,dir,v) representing a gen decision s at valence v, for head word h in direction dir, the update equation for the gen distribution was:

$$\hat{P}^{n+1}_{gen}(g|h,dir,v) = \frac{E^n(g,h,dir,v)}{E^n(\text{Bound},h,dir,v) + E^n(\text{Dep},h,dir,v))}$$

6.2.2 The DMV with Backoff

Headden et al. (2009) elaborated the DMV to incorporate multiple sources of information per word. While the original DMV learned from POS tags, Headden et al. (2009) presented a version which learns from words and POS tags. Intuitively, the new version pays more attention to word information when it has a lot of evidence for that

word, and pays more attention to POS tag information when there is little evidence for the word. Headden et al. present several additions to the DMV, including an additional conditioning term in P_{choose} and an intensive initialization scheme that involves running the training procedure from hundreds of random initializations. In this section, however, we describe only their implementation of backoff with the original DMV.

Technically, Headden et al. (2009) borrow interpolated backoff methods from language modeling with n-grams. When using n-grams, there is a trade-off involved in choosing large or small n. Choosing large n allows the language model to capture long-range dependencies that small n does not. For example, because of "either ... or" constructions, there is a strong statistical expectation of seeing "or" after seeing "either," but these two words can easily be separated by several words. Picking an n of, say, 6 thus allows the language model to capture this construction any time it occurs with up to four intervening words, while an n of 3 will almost never capture it. However, picking large n also means that the language model is likely to be very sparse: each possible sequence of length $n = 6$ will occur many fewer times than each possible sequence of length $n = 3$. Choosing large n can accordingly lead to a model with very poor parameter estimates. For example, most determiners appear very near their nouns. A language model with $n = 3$ will be able to capture this short-range dependency with a relatively small number of parameters: at most the number of determiners times the number of observed nouns (perhaps with an Out-Of-Vocabulary token). A language model with $n = 6$, however, will count each determiner-noun sequence as distinct if they all appear in different contexts.

Interpolated backoff (e.g. Chen and Goodman, 1996) provides a solution for this trade-off by taking a weighted average of different n-grams. Suppose we would like to use trigrams, backing off to bigrams. Interpolated backoff estimates the trigram probability $\hat{P}$ as:

$$\hat{P}(w_n|w_{n-1},w_{n-2}) = \lambda P(w_n|w_{n-1},w_{n-2}) + (1-\lambda)P(w_n|w_{n-1})$$

with $\lambda \in [0,1]$. λ can be tuned on a dev set to be large when there are many counts of $w_{n-1}w_{n-2}$, and small when $w_{n-1}w_{n-2}$ is rare. Throughout this chapter, the $\hat{}$ diacritic is used to indicate that a distribution is an estimate.

Headden et al. (2009) similarly build a model that takes a weighted average of distributions that condition on different kinds of information. In the n-gram example, λ is a scalar parameter. However, since it is used to specify two events (backing off or not backing off), and the values associated with each event sum to 1, it actually specifies

a probability distribution over the decision to backoff or not backoff. Since we will be learning different backoff probabilities depending on the head, attachment direction, and valence, the backoff decision will be explicitly written as a probability distribution. To remember that this probability distribution is over backoff distributions, we will re-use the λ symbol instead of P. Specifically, $\lambda_{gen}(\cdot)$ will represent our probability distribution over backing off in the Gen decision, and $\lambda_{choose}(\cdot)$ will represent our probability distribution over backing off in the Choose decision.

Using h_p to represent head POS tag and h_w to represent head word, one of the models Headden et al. explored estimates:

$$\begin{aligned}\hat{P}_{gen}(\cdot|h_w,h_p,dir,val) =& \lambda_{gen}(\neg\text{Backoff}|h_w,h_p,val,dir)P_{gen}(\cdot|h_w,h_p,dir,val)+ \\ & \lambda_{gen}(\text{Backoff}|h_w,h_p,val,dir)P_{gen}(\cdot|h_p,dir,val) \end{aligned} \tag{6.1}$$

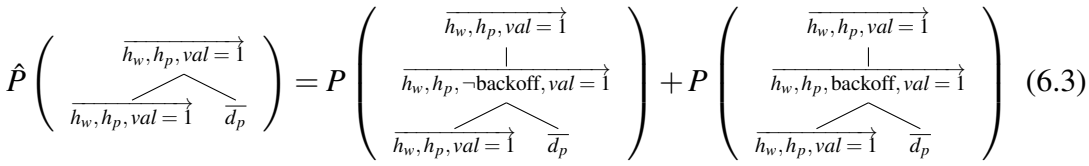
Concretely, the first, non-backoff P_{gen} term on the right hand side is a (sparse but fine-grained) probability distribution that decides whether to generate a Boundary or Dependent based on head word, head POS tag, valence, and direction, while the second, backoff P_{gen} has marginalized out the head word information, and decides whether to generate a Boundary or a Dependent based on only the head POS tag, valence, and direction. Each P_{gen} distribution is weighted by the corresponding Backoff probability. Note that $\lambda_{gen}(\neg\text{Backoff}|\cdot)$ and $\lambda_{gen}(\text{Backoff}|\cdot)$ both condition on head word.

And using d_p to represent dependent POS tag:[2]

$$\begin{aligned}\hat{P}_{choose}(d_p|h_w,h_p,dir) =& \lambda_{choose}(\neg\text{Backoff}|h_w,h_p,dir)P_{choose}(d_p|h_w,h_p,dir)+ \\ & \lambda_{choose}(\text{Backoff}|h_w,h_p,dir)P_{choose}(d_p|h_p,dir) \end{aligned} \tag{6.2}$$

For simplicity, the backoff decision can be viewed as an extra unary rule, which decorates its child with an annotation representing the decision to backoff or not to backoff. Accordingly, we can express the Choose decision of Equation 6.2 in probabilities of subtrees (assuming the dependent is to the right of the head, and this is not the first rightward dependency for this head):

[2] Headden et al. (2009) also condition on *val* in P_{choose} and λ_{choose}, in their elaboration of the DMV called the Extended Valence Grammar (EVG), but we do not and so for clarity omit that conditioning event.



(6.3)

The top-most rule in each tree on the right hand side of Equation 6.3 represents the decision to backoff, and corresponds to (i.e. occurs with probability) the λ_{choose} terms in Equation 6.2. Similarly, the bottom rule in each tree on the right hand side of Equation 6.3 corresponds to the Choose probability from either the full distribution or the backoff distribution represented by the respective P_{choose} on the right hand side of Equation 6.2.

Note that the rightmost tree retains the h_w annotation even though its corresponding P_{choose} term in Equation 6.2 does not refer to h_w. Headden et al. (2009) point out that the ignored information can be preserved in the tree by *tying* backed-off rules together. Rules are tied by defining a *tying relation* among non-terminals which should have the same probability for each expansion. In this case, we tie non-terminals which have the same head POS tag but different head words. The probabilities are drawn from a multinomial indexed by the tying relation (here, POS tag and direction of attachment) rather than the non-terminal itself.

To obtain good performance, Headden et al. (2009) impose an UNK cutoff c. In a pre-processing stage, any word which appears less than c times in the training data is replaced with a special UNK token. For best performance, they use a cutoff of 100.

6.2.2.1 Variational Bayes for the DMV with Backoff

Headden et al. (2009) train all models using Variational Bayes with Dirichlet priors. This section provides an intuitive overview of Variational Bayes for the DMV; for details, see Kurihara and Sato (2006).

The original DMV was trained with Expectation Maximization, which performs maximum-likelihood estimation (MLE), finding the model parameters $\hat{\theta}$ that maximize the probability of our corpus C, averaging over all possible dependency trees $\mathbf{t}$:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} P(C|\theta) = \underset{\theta}{\operatorname{argmax}} \sum_{\mathbf{t}} P(\mathbf{t}, C|\theta)$$

Maximum likelihood estimates can suffer in the face of small data, however. For example, they perform poorly with rare events, allocating extra probability mass to those

rare events that happen to have occured in the training data while giving no probability mass to rare events which happen to be absent from the training data. MLE is particularly inappropriate for the DMV with Backoff. First, since the model has a lexicalized component, there will be many rare events, meaning the model will have many opportunities to overestimate the probability of rare events which occur in the training corpus, and many opportunities to underestimate the probability of rare events which do not occur in the training corpus. Second, it will give zero probability to the backoff decision, because backing off, by definition, reserves probability mass from the observed data for unobserved events (i.e. unseen events that also condition on the backoff set).

The first problem could, of course, be addressed in a maximum-likelihood framework by applying any of the standard frequentist smoothing approaches. The second problem could also be addressed in a maximum likelihood framework by holding out additional training data, fitting the usual DMV parameters on the resulting training set, and then holding the DMV parameters fixed while tuning the backoff λ parameters on the held-out data set. However, this approach is unappealing because it further reduces the amount of data available for estimating our P_{root}, P_{choose}, P_{gen}, and λ probability distributions.

A Bayesian approach provides a more principled solution for these problems while also allowing us to use all of our data in estimating all of our probability distributions. There are two primary advantages to using a Bayesian approach: model averaging, and the ability to set prior probabilities on model parameters. Model averaging automatically addresses the first problem: instead of picking a single "best" model (which likely overallocates probability mass to those rare events we happen to have observed), we will get the average answer from all models. Those models which overcommit to rare events will have a low probability given the data (even though the probability of the data given the model is maximized by such models), and so make a much smaller contribution to the final distribution over parse trees. To solve the second problem, we can select priors for our λ distributions that prefer the backoff distribution when the extra information is rarely observed (and so we have poor evidence for the extra stream), and prefer the joint distribution when the extra information is often observed (and so we have good evidence for the extra stream).

Specifically, instead of picking one set of parameters $\hat{\theta}$ that maximizes the probability of the observed corpus C (i.e. maximizes $P\left(C|\hat{\theta}\right)$), Bayesian approaches estimate a distribution over θ and the hidden variables $\mathbf{t}$ given the data and the prior specification

α: $P(\mathbf{t}, \theta | C, \alpha)$. As discussed in Section 3.3.2, this distribution can be better understood by applying Bayes' rule:

$$P(\mathbf{t}, \theta | C, \alpha) = \frac{P(C, \mathbf{t}|\theta)P(\theta|\alpha)}{\int_{\Delta\theta} \sum_{\mathbf{t}} P(C, \mathbf{t}|\theta)P(\theta|\alpha)d\theta} \tag{6.4}$$

Bayes' rule allows us to express this joint distribution in terms of the likelihood function $P(C, \mathbf{t}|\theta)$, which will just be a variant of the DMV, and a prior $P(\theta|\alpha)$. However, because it involves computing a sum over all possible treees for all possible θ, this joint distribution is difficult to compute.

Variational Bayes is a class of approaches to approximating this distribution. The general idea is to pick a constrained family of distributions that are easy to compute, and learn a distribution $Q(\mathbf{t}, \theta)$ in this family of distributions that most closely approximates the true posterior. Headden et al. (2009) use Mean Field Variational Bayes, which assumes that the variational distribution $Q(\mathbf{t}, \theta)$ factors into $Q(\mathbf{t})Q(\theta)$. Intuitively, this independence assumption allows us to find one distribution over model parameters $Q(\theta)$, and another over our hidden variables $Q(\mathbf{t})$ (which, in this case, is a distribution over projective dependency trees). Computationally, this independence assumption allows mean-field VB to employ an EM-like algorithm. Specifically, VB-EM holds $Q(\theta)$ fixed while bringing the variational distribution over $\mathbf{t}$ closer to the true posterior over $\mathbf{t}$ (in a variational E-step), and then holds $Q(\mathbf{t})$ fixed while bringing the variational distribution over θ closer to the true posterior over θ (in a variational M-step). This process brings the variational distribution closer to the true posterior in the sense of minimizing the Kullback-Leibler divergence of the variational distribution from the true posterior.

Headden et al. (2009) use Dirichlet priors. Allocating some probability mass to every model is a pre-requisite for effective model averaging, and it is easy to pick Dirichlet priors that have this property. Specifically, a Dirichlet prior for a multinomial over n outcomes is defined by a vector of non-negative hyperparameters α of length n. A larger magnitude for hyperparameter i, relative to the other hyperparameters, expresses a prior bias for larger probability for outcome i. Accordingly, some probability mass can be allocated to each model by simply setting all Dirichlet hyperparameters greater than 0. Similarly, strong prior assumptions can be avoided simply by setting all hyperparameters to the same value; such a Dirichlet prior is called *symmetric*.

The training procedure for Variational Bayes with Dirichlet priors is an iterative procedure similar to EM. After initialization, expected counts $E(r_i)$ are gathered for each rule r_i using the Inside-Outside algorithm. This variational E-Step is identical

to the E-step of traditional EM. The Maximization step is similar to the M-Step of EM, differing in two ways. First, the expected counts for a rule r_i are incremented by the hyperparameter α_i for that rule. Second, the numerator and denominator are passed through a scaling function $\exp(\psi(\cdot))$. The ψ function is the *digamma* function, which is the derivative of the log gamma function (and the gamma function is in turn a generalization of the factorial function to all real numbers).

By examining specific update equations, we can see how this approach circumvents the problems encountered by MLE as mentioned above. First, let's examine the update equation for P_{choose} from iteration n to $n+1$:

$$\hat{P}^{n+1}_{choose}(d_w|h_w,dir) = \frac{\exp(\psi(E^n(r_{d_w,h_w,dir}) + \alpha_{d_w,h_w,dir}))}{\exp(\psi(\sum_c(E^n(r_{c,h_w,dir}) + \alpha_{c,h_w,dir})))}$$

We can see that, in both the numerator and denominator, summed expectations are augmented by Dirichlet hyperparameters, and then run through the $\exp(\psi(\cdot))$ function. The smoothing effect of the Dirichlet priors is apparent; in fact, aside from the $\exp(\psi(\cdot))$ expressions, the formula looks exactly like add-λ smoothing.

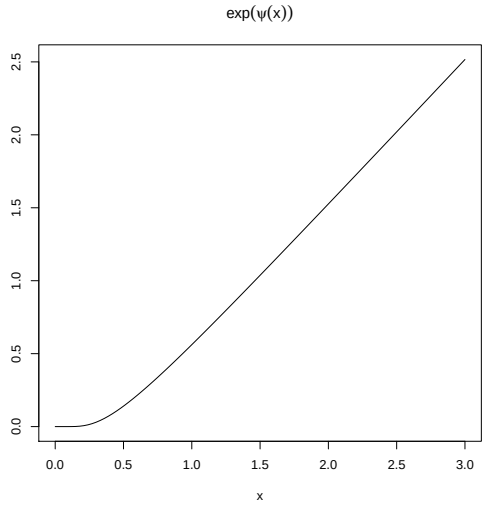


Figure 6.4: $\exp(\psi(\cdot))$ function from 0 to 3

Figure 6.4 presents a plot of the $\exp(\psi(\cdot))$ function, showing that it is approximately the same as subtracting 0.5 from the input for input larger than about 2. Concretely, the $\exp(\psi(\cdot))$ function prevents us from believing very small counts too much: we could have just as easily seen other rare events instead. The model averaging, then, is implemented by this $\exp(\psi(\cdot))$ function.

Now let's examine the update equations for λ_{choose}. Since there are only two possible decisions, ¬Backoff and Backoff, we'll explicitly write out both equations instead of using a variable:

$$\hat{\lambda}^{n+1}_{choose}(\neg\text{Backoff}|h_w, dir) = \frac{\exp(\psi(\alpha_{\neg\text{Backoff}} + \sum_c(E^n(r_{c,h_w,dir}))))}{\exp(\psi(\alpha_{\text{Backoff}} + \alpha_{\neg\text{Backoff}} + \sum_c(E^n(r_{c,h_w,dir}))))}$$

$$\hat{\lambda}^{n+1}_{choose}(\text{Backoff}|h_w, dir) = \frac{\exp(\psi(\alpha_{\text{Backoff}}))}{\exp(\psi(\alpha_{\text{Backoff}} + \alpha_{\neg\text{Backoff}} + \sum_c(E^n(r_{c,h_w,dir}))))}$$

Since only the ¬Backoff term includes the expected counts, as we see h_w in direction *dir* more frequently, the ¬Backoff term will swamp the Backoff term. By picking α_{Backoff} to be larger than $\alpha_{\neg\text{Backoff}}$, we can bias our λ distribution to prefer backing-off until we've seen at least $\alpha_{\text{Backoff}} - \alpha_{\neg\text{Backoff}}$ (expected) arcs out of h_w in the direction of *dir*. Moreover, by increasing the absolute value of our hyperparameters, we can slow down how quickly λ switches from mostly preferring the Backoff distribution to mostly preferring the ¬Backoff distribution.

Finally, as both EM and VBEM are greedy gradient traversal methods, they are only able to find a *local* optimum for their objective. As the objectives are complex for natural language data, EM and VBEM are both sensitive to initialization. One way to address this is to engineer an initialization that is close to a good local optimum. Klein and Manning (2004) take this approach in their original formulation of the DMV. Another approach is to try several random initializations, and select the initialization that results in the best objective score. Headden et al. (2009) take this second approach, running 20 random initializations and picking the best run in terms of objective score out of the 20. Their final performance scores are obtained by running 50 sets of random initializations, and averaging the performance of the best run of each of the 50 sets. However, the focus in our experiments is the *relative* performance of our parsing models when learning from different sorts of input. Accordingly, these experiments will use the same sort of harmonic initialization procedure used in Klein and Manning (2004) (the specific equations will be provided in Section 6.3.2).

6.2.3 Predictability DMV

The models explored in this chapter are straightforward applications of the DMV with Backoff to words and (quantized) word duration information. For comparison, we will also examine applications of the same DMV with Backoff that see words and either Part-of-Speech (POS) tags or Break Index. Backoff models are all presented with two

streams of information (providing two of word identity, POS tag, or word duration at each time step). One stream will be called the "backoff" stream, and the other stream will be called the "extra" stream. The model will learn a probability distribution conditioning on the cross-product of the extra and backoff state space, backing off to a probability distribution conditioning on only the backoff stream. Our first words and duration model, for example, takes the duration as the "extra" stream and the word identity as the "backoff" stream, and, using h_a to represent the acoustic information for the head, models:

$$\begin{aligned}\hat{P}_{gen}(\cdot|h_w,h_a,dir,val) =&\lambda_{gen}(\neg\text{Backoff}|h_w,h_a,val,dir)P_{gen}(\cdot|h_w,h_a,dir,val)+\\ &\lambda_{gen}(\text{Backoff}|h_w,h_a,val,dir)P_{gen}(\cdot|h_w,dir,val)\end{aligned} \tag{6.5}$$

$$\begin{aligned}\hat{P}_{choose}(d_w|h_w,h_p,dir) =&\lambda_{choose}(\neg\text{Backoff}|h_w,h_a,dir)P_{choose}(d_w|h_w,h_a,dir)+\\ &\lambda_{choose}(\text{Backoff}|h_w,h_a,dir)P_{choose}(d_w|h_w,dir)\end{aligned} \tag{6.6}$$

This model considers the "extra" stream only for the conditioning head, and does not generate the "extra" stream for the dependents. Accordingly, this is actually a conditional, rather than fully generative, model of the observed backoff stream and unobserved syntax on the "extra" stream. For this reason, we will refer to this model as the "Cond." model in our experiments.

It may be useful to directly model the "extra" stream among the dependents as well. Among models learning from words and POS tags, modeling dependent word identity could capture such facts as that the direct object of "eat" should usually be a food item. Alternatively, among models learning from words and word durations, we would expect modeling dependent word duration to help because Gahl and Garnsey (2004) found that dependents exhibit longer word durations in low-probability frames. Accordingly, we also explore variants that generate the dependent "extra" stream along with the dependent "backoff" stream. First, we examine a model (called "Joint") that generates them jointly:

$$\hat{P}_{choose}(d_w,d_a|h_w,h_p,dir) = \tag{6.7}$$

$$\begin{aligned}&\lambda_{choose}(\neg\text{Backoff}|h_w,h_a,dir)P_{choose}(d_w,d_a|h_w,h_a,dir)+\\ &\lambda_{choose}(\text{Backoff}|h_w,h_a,dir)P_{choose}(d_w,d_a|h_w,dir)\end{aligned} \tag{6.8}$$

However, a fully-joint model will have a very large state-space and may suffer from data sparsity. To reduce the number of parameters, we also explore a model (called "Indep.") that generates the "extra" and "backoff" streams independently, given the

head information and attachment direction.

$$\begin{aligned}\hat{P}_{choose}(d_w, d_a | h_w, h_p, dir) = \\ \lambda_{choose}(\neg\text{Backoff} | h_w, h_a, dir) P_{choose\_backoff}(d_w | h_w, h_a, dir) P_{choose\_extra}(d_a | h_w, h_a, dir) + \\ \lambda_{choose}(\text{Backoff} | h_w, h_a, dir) P_{choose\_backoff}(d_w | h_w, dir) P_{choose\_extra}(d_a | h_w, dir) \end{aligned} \quad (6.9)$$

We also modified the DMV with Backoff slightly to handle heavily lexicalized models better. Since expectations for estimating the P_{choose} distribution are computed separately for each sentence, arcs between words which never appear in the same sentence in the training data get zero probability if treated naïvely. In the original DMV with Backoff, as formulated by Headden et al. (2009), arcs between such words are given probability mass only by virtue of the backoff distribution to POS tags, which all appear in the same sentence at least once. As our focus is on language acquisition, however, the most interesting models will not have gold-standard POS tags. To allow arcs between words which never appear in the same sentence to receive a probability greater than zero, we add one extra $\alpha_{\texttt{UNK}}$ hyperparameter to the Dirichlet prior of P_{choose} for each combination of conditioning events. This hyperparameter reserves probability mass for a word h_w to take a word d_w as a dependent if h_w and d_w never appeared together in the training data. Specifically, the update equation for P_{choose} is now:

$$\hat{P}^{n+1}_{choose}(d_w | h_w, dir) = \frac{\exp(\psi(E^n(r_{d_w,h_w,dir}) + \alpha_{d_w,h_w,dir}))}{\exp(\psi(\alpha_{\texttt{UNK},h_w,dir} + \sum_c (E^n(r_{c,h_w,dir}) + \alpha_{c,h_w,dir})))}$$

So if we are considering d_w as a dependent for h_w, and h_w and d_w never appeared in the same sentence in the training data (in the direction of *dir*), h_w takes d_w as a dependent with probability:

$$\hat{P}_{choose}(d_w | h_w, dir) = \frac{\exp(\psi(\alpha_{\texttt{UNK},h_w,dir}))}{\exp(\psi(\alpha_{\texttt{UNK},h_w,dir} + \sum_c (E^{convergence}(r_{c,h_w,dir}) + \alpha_{c,h_w,dir})))}$$

Essentially, this maintains an $\texttt{UNK}_{h_w,dir}$ token specific to each head word and direction. This token is never observed in the training data; at evaluation time, if we encounter a putative dependent that never appeared with the putative head, we map it to $\texttt{UNK}_{h_w,dir}$ for one probability look-up. The amount of probability mass reserved for unseen dependents will decrease as we see h_w more often.

Note that this is very different from the global UNK cutoff used by Headden et al. (2009). The global UNK cutoff is imposed in a preprocessing step, and so affects every

occurrence of an UNK'd word in all our P and λ probability tables. This use of $\alpha_{\texttt{UNK}}$, on the other hand, affects only dependents in P_{choose}, and treats a dependent as UNK iff it did not occur on that particular side of that particular head word in any sentence. We will end up using both global UNK cutoffs (optimized on the dev set) and these $\alpha_{\texttt{UNK},h_w,dir}$ hyperparameters.

Finally, our P_{root} distribution ignores the "extra" stream in the Cond. model, considers the joint distribution over the "extra" and "backoff" stream in the Joint model, and assumes the "extra" and "backoff" streams are independent in the Indep. model.

6.3 Experiments and Results

6.3.1 Datasets

We evaluate our models on three datasets. First, we evaluate on sentences from the Wall Street Journal portion of the Penn Treebank with ten words or fewer (`wsj10`) in order to validate our variant of the DMV with Backoff on the same datset used by Headden et al. (2009). Second, we evaluate on a larger portion of the Switchboard dataset from Chapter 5. Third, we evaluate on a portion of the Brent corpus of child-directed speech (Brent and Siskind, 2001).

6.3.1.1 `wsj10`

We evaluate our version of the DMV with Backoff on `wsj10`, which does not have any word duration or break index information, for two reasons. First, we are using new formulations of the DMV with Backoff. Headden et al. (2009) don't give probability mass to words that never appear in the same sentence, but we do with $\alpha_{\texttt{UNK}}$. Additionally, Headden et al. consider only the conditional formulation of the DMV with Backoff, while we also evaluate fully generative versions. Evaluating on the same dataset will allow us to verify that this variant still behaves sensibly on a standard dataset, and so should be a good basis for exploring child-directed speech. Second, Headden et al. (2009) use an intensive initializer that relies on hundreds of random restarts, and so, strictly speaking, only show that the backoff technology is useful for good initializations. Our new evaluation, as we will see, shows that the backoff technology provides a substantial benefit even for just the harmonic initialization.

As is standard, `wsj10` was created by removing all punctuation and traces from sentences of the Wall Street Journal portion of the Penn Treebank. Sentences containing

	train	dev	test
Word tokens	42,505	1,765	2,571
Word types	7,804	818	1,134
Sentences	6,007	233	357

Table 6.1: `wsj10` dataset figures.

more than ten tokens after this removal were then discarded. Input containing words was lowercased, and all numbers were replaced with the token "NUMBER." Following Headden et al. (2009), we used sections 2 through 21 as a training set, section 22 as a development set, and section 23 as a test set. Table 6.1 presents final corpus statistics.

`wsj10` is drawn from the Wall Street Journal portion of the Penn Treebank, which contains hand-annotation of constituency parses, not dependency parses. As is standard, we used Johansson and Nugues's (2007) "constituent-to-dependency" conversion tool to obtain high-quality CoNLL-style dependency parses.

6.3.1.2 `swbdnxt`

We also evaluate all systems on the same `swbdnxt` corpus used in Chapter 5. This evaluation serves three primary purposes. First, as this corpus contains the same kind of annotation as the `wsj10` corpus just described, and consists of conversational, adult-directed speech, it will provide a fairly direct comparison for unsupervised parsing of spontaneous speech and unsupervised parsing of edited text. Second, it will allow us to compare the relative utility of word duration with hand-annotated, gold-standard POS tags. Third, as our models do successfully learn from word durations, this evaluation will provide an important replication of the basic result from Chapter 5 with a very different kind of syntactic model.

As with the `wsj10` dataset, `swbdnxt` is annotated with only constituency parses, but we would like to evaluate dependency parses. To provide an approximate "gold-standard," the same constituency-to-dependency conversion tool was run on the development and test sets. 200 sentences of this gold standard were randomly selected to evaluate the accuracy of the conversion tool. Excluding arcs involving words that play no clear role in syntactic dependency structure (such as "um"), about 86% of the arcs were correct. Although this rate is uncomfortably low, it is still a much higher accuracy rate than unsupervised dependency parsers typically achieve. Accordingly, it probably provides a reasonable measure of *relative* dependency parse quality among

	Train	Dev	Test
Word tokens	24,998	7,981	8,746
Word types	2,647	1,459	1,548
Sentences	3,998	778	802

Table 6.2: `swbdnxt` statistics

competing systems. Accordingly, the output of the converter tool was used as the gold standard. Finally, all sentences longer than ten words were thrown out of the training set. Table 6.2 presents final corpus statistics.

Notice that the dev and test sets are identical for our `swbdnxt` experiments and the chunking experiments from Chapter 5; only the training set is varied. Also, unlike `wsj10`, we evaluate on all sentences longer than length three, not only those of ten words or less.

6.3.1.3 `Large Brent`

In the evaluation of most interest, we evaluated our models on a portion of the Brent corpus of Child Directed Speech (Brent and Siskind, 2001). Specifically, we used the `Large Brent` dataset introduced in Rytting et al. (2010). `Large Brent` consists of utterances from four of the mothers in Brent and Siskind (2001), and provides a forced-alignment of a dictionary-based phonetic transcription to the audio itself. This forced alignment will provide our word duration measures. Rytting et al. (2010) divide `Large Brent` into a 90%/10% train/test partition. To avoid overfitting, we extract every ninth utterance from the original training partition to create a development set, producing an 80%/10%/10% partition. Finally, as we are interested in recovering grammatical dependencies such as subjecthood and negation, clitics were separated from their base word. While it is traditional in unsupervised dependency parsing to exclude long sentences, the longest sentence in the dataset had only 22 words, with the vast majority shorter than 17 words. Spitkovsky et al. (2010) investigated the influence of training sentence length on DMV performance, and reported that the DMV is capable of using training sentences longer than 10 (the cutoff in `wsj10`). Specifically, Spitkovsky et al. reported that including sentences of up to 15 to 20 words in length in the training data led to improvement (depending on the exact comparison), but including sentences much longer than about 20 words could lead to a degradation in performance. Accord-

	Train	Dev	Test
Word tokens	20,954	2,127	2,206
Word types	1,390	482	488
Sentences	6,249	424	449

Table 6.3: `Large Brent` dataset figures.

ingly, no sentences are thrown out.[3] Corpus statistics are presented in Table 6.3.

The original Brent corpus is distributed through CHILDES (MacWhinney, 2000) with dependency annotations. However, these are not hand-corrected, and rely on a different tokenization of the dataset than is present on the word tier. To produce a reliable gold-standard, we annotated all sentences of length 2 or greater from the development and test sets with dependencies drawn from the Stanford Typed Dependency set (de Marneffe and Manning, 2008) using the annotation tool used for the Copenhagen Dependency Treebank (Kromann, 2003).

6.3.1.4 Acoustic Information

We are hoping to learn a model over observed word duration and hidden syntax. We expect this to work because, as previously discussed, words in high-probability structures tend to be pronounced quickly, and words in low-probability structures tend to be pronounced slowly. As a very simple model of word duration, then, we will simply classify a word as in the longest third duration (hyper-articulated), shortest third (hypo-articulated), or middle third. Since one of our models makes an independence assumption between dependent word identity and dependent word duration, we would like to eliminate influences on word duration due to basic word form. Accordingly, the classification is done on the basis of vowel count: each word with 0 vowels is classified as in the shortest, longest, or middle tercile of duration among words with 0 vowels, each word with 1 vowel is classified with respect to terciles computed among words with 1 vowel, and so on. There was only one word token with five vowels, and this word was included among the four-vowel words for the purposes of duration tercile. We also tried normalizing for talker identity in the same way, and obtained comparable results. Finally, quintiles rather than terciles were also explored, but they yielded slightly lower dev-set performance.

[3] We note the interesting correspondence between sentence length distributions that are useful for the DMV and sentence length distributions that are present in at least this corpus of child-directed Speech.

6.3.1.5 Difficulty of the task

Before proceeding to experiments with our models, let's consider the task under a very simple model.[4] Specifically, the unlabeled dependency parses we wish to produce can be represented as a list of numbers, with each number encoding the directed distance to that word's parent. For example, the sentence "I see you" would receive the dependency parse $(1, 0, -1)$, because the parent of the first word is one word to the right, the second word is the root, and the parent of "you" is one word to the left.

As an easy-to-compute measure of the difficulty of the unlabeled dependency parsing task, we can consider the entropy of the probability distribution over these directed distances. A dependency parse in this model is thus viewed as a sequence of independent draws from this probability distribution. This model knows only the relative frequency of directed distances, and does not know the words, that the arcs must form a tree, or even how long each sentence is (which limits the lengths of the arcs of each sentence).

The entropy of the probability distribution over directed distances on the test set of `swbdnxt10` is 3.16 bits, and the entropy of the distribution over directed distances on the `Large Brent` test set is 2.69 bits. This means, as we might expect, that parsing adult-directed speech is harder (although this difference might also reflect the fact that the adult-directed parses have been automatically obtained from constituency structure). Concretely, this means that our simple model has a perplexity of $2^{3.16} = 8.9$ on the adult-directed speech: it chooses between 8.9 effective directed distances. It has a perplexity of $2^{2.69} = 6.4$ on the child-directed speech.

We can also examine the conditional entropy of directed distances given annotated break index or word duration. This quantity tells us how uncertain we are about each directed distance after we already know the break index or the word duration, under this simple model. If break index or word duration tell us everything we need to know, the conditional entropy will be zero, and if they tell us nothing, the conditional entropy of the directed distances will be the same as the entropy of the directed distances.

The directed distance conditional entropy given break index on `swbdnxt10` is 3.05, for a perplexity of 8.3, eliminating 0.6 effective alternatives. The directed distance conditional entropy given duration tercile on `swbdnxt10` is 3.09, for a perplexity of 8.5, ruling out 0.4 effective alternatives. For `Large Brent`, the directed distance conditional entropy given duration tercile is 2.58, for a perplexity of 6.0, eliminat-

[4] Thanks go to Mark Liberman for prompting this investigation.

ing 0.4 effective alternatives. Evidently, neither word duration nor annotated break index are very informative about unlabeled dependency parsing under this extremely simple model.

We can interpret this result using the statistical dependency-centric view of bootstrapping accounts introduced in Chapter 3. Our very simple model here assumes that the functional form of the statistical dependency between word duration and unlabeled dependency parsing, or between break index and unlabeled dependency parsing, refers only to each word token in a string in isolation, and specifically does not refer to lexical identity, part-of-speech, statistical dependencies between word tokens of the same utterance, or a tree-structure constraint for each sentence. The failure of this model to uncover useful statistical dependencies does not necessarily mean word duration or break index do not co-vary with syntactic dependencies in a useful way; rather, it means that this assumed functional form is ill-suited.

Next, we will turn to experiments with our DMV-based models of words, dependency syntax, and either word duration or break index. These models provide much more complex functional forms for potential statistical dependencies between word duration and dependency syntax, or break index and dependency syntax. In these experiments, we will see if this assumed functional form can adequately capture the covariance of these variables in the data, and to what extent the specific shape of the particular statistical dependency reflects prosodic structure or predictability effects.

6.3.2 Initialization

As previously mentioned, we will follow Klein and Manning (2004) in using a *harmonic* initializer. Broadly, a harmonic initializer seeks to initialize the grammar with counts that give heavier weight to arcs between words that are often close together. Specifically, we set our initial expected counts E^0 according to the following equations, and then run the corresponding M-Step to obtain our initial grammar.

The initial Choose expected count for each pair of terminals depends on how often they appear in the same sentence and how close they are:

$$E^0_{choose}(d,h,\rightarrow) = \sum_{s \in C} \left(\sum_{i=0}^{\text{len}(s)-2} \left((s(i) == h) \sum_{j=i+1}^{\text{len}(s)-1} \left((s(j) == d) + \frac{(s(j) == d)}{(j-i)} \right) \right) \right)$$

$$E^0_{choose}(d,h,\leftarrow) = \sum_{s \in C} \left(\sum_{j=\text{len}(s)-1}^{1} \left((s(j) == h) \sum_{i=j-1}^{0} \left((s(i) == d) + \frac{(s(i) == d)}{(j-i)} \right) \right) \right)$$

for sentences s in corpus C, and infix $==$ s.t. an expression $s(n) == w$ returns 1 if the observation (whether that observation is a word, a POS tag, a word duration, or a tuple of any of these) at 0-based position n of sentence s is w, and returns 0 otherwise.

The initial Bound expected count of an observation h with $val = T$ in a direction *dir* is incremented if the sentence has no more words in the direction of *dir*, and Dep is incremented otherwise:

$$E^0_{gen}(\text{Bound}, h, \rightarrow, val = T) = 1 + \sum_{s \in C} (s(\text{len}(s) - 1) == h)$$

$$E^0_{gen}(\text{Dep}, h, \rightarrow, val = T) = 1 + \sum_{s \in C} \left(\sum_{i=0}^{\text{len}(s)-2} s(i) == h \right)$$

$$E^0_{gen}(\text{Bound}, h, \leftarrow, val = T) = 1 + \sum_{s \in C} (s(0) == h)$$

$$E^0_{gen}(\text{Dep}, h, \leftarrow, val = T) = 1 + \sum_{s \in C} \left(\sum_{i=1}^{\text{len}(s)-1} s(i) == h \right)$$

The initial Bound expected count of an observation h with $val = F$ in a direction *dir* is incremented if the sentence has exactly one more word in the direction of *dir*, and Dep is incremented otherwise:

$$E^0_{gen}(\text{Bound}, h, \rightarrow, val = F) = 1 + \sum_{s \in C} (s(\text{len}(s) - 2) == h)$$

$$E^0_{gen}(\text{Dep}, h, \rightarrow, val = F) = 1 + \sum_{s \in C} \left(\sum_{i=0}^{\text{len}(s)-3} s(i) == h \right) + (s(\text{len}(s) - 1) == h)$$

$$E^0_{gen}(\text{Bound}, h, \leftarrow, val = F) = 1 + \sum_{s \in C} (s(1) == h)$$

$$E^0_{gen}(\text{Dep}, h, \leftarrow, val = F) = 1 + \sum_{s \in C} \left(\sum_{i=2}^{\text{len}(s)-1} s(i) == h \right) + (s(0) == h)$$

Finally, the initial expected count for Root is just incremented according to the frequency of the word:

$$E^0_{root}(h) = \sum_{s \in C} \left(\sum_{i=0}^{\text{len}(s)-1} (s(i) == h) \right)$$

6.3.3 Parameters

In all experiments, hyperparameters for P_{root}, P_{gen}, and P_{choose} (along with associated backed-off distributions, and including $\alpha_{\texttt{UNK}}$) were 1, α_{Backoff} was 10, and $\alpha_{\neg\text{Backoff}}$ was 1. Variational Bayes EM was run on the training set until the data log-likelihood changed by less than 0.001%, and then the parameters were held fixed and used to obtain Viterbi parses for the evaluation sentences. Finally, we explored different global UNK cutoffs, replacing each word that appeared less than c times with the token `UNK`. Note that this is very different from $\alpha_{\texttt{UNK}}$, as $\alpha_{\texttt{UNK}}$ affects only dependents in P_{choose} and has different effects for each head word and each direction. We ran each model 5 times (with the same initialization procedure), once each for $c \in \{0, 1, 25, 50, 100\}$. We picked the c which scored best on the development set for running on the test set and presentation here.

6.3.4 Evaluation

In addition to evaluating the various incarnations of the DMV with backoff and input types, we compare to uniform branching baselines, the Common Cover Link (CCL) parser of Seginer (2007), and the Unsupervised Partial Parser (UPP) of Ponvert et al. (2011). The UPP produces a constituency parse from words and punctuation using a series of finite-state chunkers; we use the best-performing (Probabilistic Right Linear Grammar) version. The CCL parser produces a constituency parse using a novel "Cover Link" representation, scoring these links heuristically. Both CCL and UPP rely on punctuation (though according to Ponvert et al. (2011), UPP less so), which our input is missing. The left-headed "LH" baseline assigns the first word of every sentence to be the root, and assumes each word takes the word immediately to its right as a dependent. As a constituency baseline, this comes down to a uniform-right branching assumption. The right-headed "RH" baseline, conversely, assigns the last word of every sentence to be the root, and assumes that each word takes the word immediately to its left as a dependent. As a constituency baseline, this is just a uniform left-branching assumption.

We evaluate the output of all models in terms of both constituency scores and dependency accuracy. We evaluate on both sorts of syntactic structure for two reasons, one theoretical and one practical. Theoretically, since dependency grammar does not focus so heavily on modeling particular word orders, it is broadly agreed that dependency grammar extends more naturally to languages with relatively free word order

(e.g. Nivre et al., 2007) and so is more cross-linguistically valid. If an unsupervised approach to learning dependency structure provides a good basis for bootstrapping constituency structure, then infants might initially look for dependencies, emphasizing constituency structure only as they find stronger evidence for it in their particular language. Practically, our `wsj10` and `swbdnxt10` corpora are originally annotated for constituency structure, with the dependency gold standard automatically derived from that structure, while our `Large Brent` corpus is originally annotated for dependency structure, with the constituency gold standard automatically derived from the dependency structure. Evaluating on both constituencies and dependencies gives us one evaluation against hand-annotated structure for each experiment.

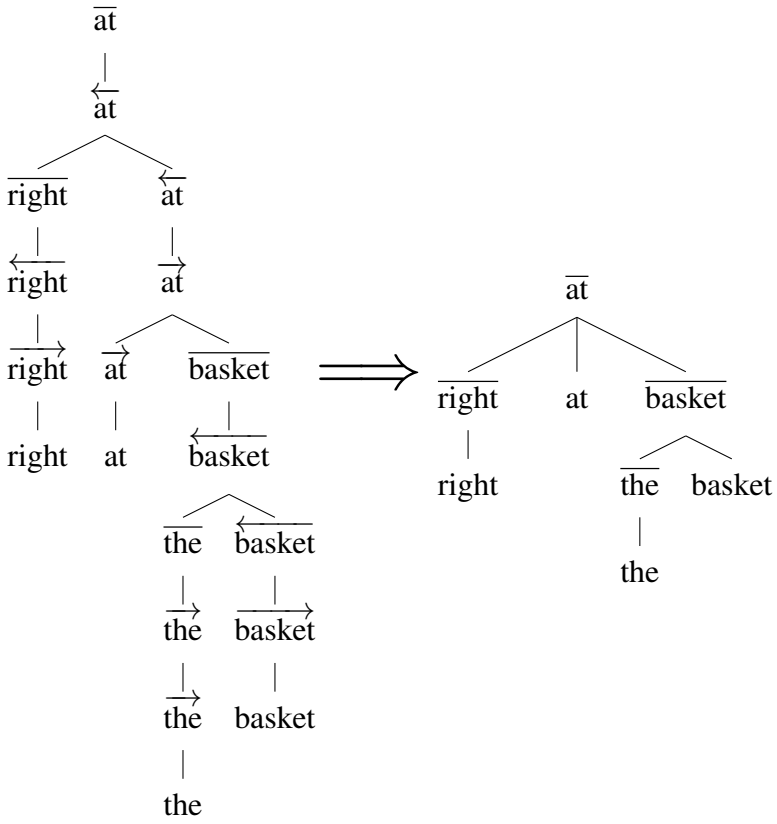


Figure 6.5: Full DMV Constituency Tree to order-agnostic DMV Constituency Tree

For constituency scores, we present the standard unlabeled Precision, Recall, and F-measure scores. To evaluate our dependency parses as constituency parses, we define a constituent to span a head and each of its dependents. Figure 6.5 shows how we can extract such constituents from the PCFG representation of the DMV by simply deleting nodes decorated with an arrow (note in particular that this transformation

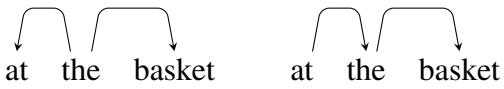


Figure 6.6: Defensible analyses for nouns and determiners

eliminates any influence of our decision to take dependents to the left before dependents to the right). We ignored constituents in both the gold standard and proposed parses that contained only one word, as such "constituents" are simply words that took no dependents.

For dependency scores, we present Directed attachment accuracy, Undirected attachment accuracy, as well as the "Neutral Edge Detection" (NED) score introduced by Schwartz et al. (2011). Directed attachment accuracy counts an arc as a true positive if it correctly identifies both a head and a dependent. Undirected attachment accuracy counts an arc as a true positive if it correctly identifies two words connected by an arc in the gold standard, but does not require that arc to correctly identify which word is the head and which is the dependent.

NED counts an arc as a true positive if it would be a true positive under the Undirected attachment score, and also counts an arc as a true positive if the head of the proposed arc is the gold-standard grandparent of the proposed dependent. NED is motivated by a desire to avoid penalizing models for systematically making arguably valid choices that conflict with the annotated gold-standard. For example, it is well-known that the DMV tends to model determiners as the heads of nouns, rather than as the dependents of nouns. While the theoretical syntax literature is divided over which dependency structure is accurate, dependency annotation schemes tend to annotate nouns as the head of the dependent. Accordingly, the DMV is penalized for systematically proposing a structure which has currency among syntacticians. Figure 6.6 presents both analyses for "at the basket." Clearly, the directed attachment evaluation awards no true positives if one is proposed while the other is in the gold standard, and the undirected attachment awards a true positive only for the arc between "the" and "basket." NED, however, will also award a true positive for the remaining arc, as "at" is either the gold-standard grandparent or gold-standard parent for both "basket" and "the" for both analyses.

Results will be reported in the form of a table. When a model sees only one kind of information, that is expressed by writing out the abbreviation for the relevant kind of information: "Wds" for words, "POS" for Part-Of-Speech, "Dur" for word duration,

and "BI" for hand-annotated Break Index. If a model sees two kinds of information, then both abbreviations are provided together. For baseline models without backoff, the abbreviations are joined by a "×" symbol (evocative of the fact that such models are just treating the input pairs as atoms drawn from the cross-product of the two streams). For example, the naïve model that learns from words and word durations without any backoff is written as "Wds×Dur." For the models with backoff, the abbreviations are joined by a "+" symbol (evocative of the fact that the models combine the information sources with a weighted summation). The name of the "extra" stream comes first. For example, for the DMV that backs off from conditioning on both words and POS tags to conditioning on only POS tags, the input is written as "Wds+POS."

6.3.5 Results: `wsj10`

		`wsj10`						`swbdnxt10`					
		Dependency			Constituency			Dependency			Constituency		
		UNK	Dir.	Undir.	NED	P	R	F	UNK	Dir.	Undir.	NED	P	R	F
EM	Wds	25	32.5	52.5	67.0	49.5	48.5	49.0	25	30.6	50.9	66.8	45.4	47.1	46.3
	POS	—	*46.4*	*63.8*	*78.1*	59.2	*58.1*	*58.6*	—	*53.0*	*65.0*	*76.8*	*52.5*	*52.9*	*52.7*
VB	Wds	25	29.4	52.4	70.5	51.3	52.6	52.0	25	36.1	54.9	72.7	49.0	50.0	49.5
	POS	—	43.5	61.9	77.3	*59.7*	57.1	58.4	—	51.3	62.5	74.3	47.1	46.6	46.8
Wds+POS	Cond.	50	49.9†	66.1†	79.6*	**64.2**†	**61.9**†	**63.0**†	100	45.5†	62.4†	77.8	58.4†	58.9†	58.7†
	Joint	50	46.0	63.7	79.0	62.0†	59.1	60.5*	1	49.4†	63.7	**79.6**†	60.0†	52.9	56.3†
	Indep.	25	**52.5**†	**68.0**†	**83.5**†	63.5†	61.5†	62.5†	100	**55.7**†	**65.8**	74.6†	**61.5**†	57.9†	59.6†
	LH	—	26.0	55.8	74.3	53.1	69.6	60.3	—	24.1	50.8	72.7	60.8	**82.5**	**70.0**
	RH	—	31.2	56.4	61.4	25.8	33.8	29.3	—	29.2	52.0	57.9	22.2	30.1	25.5
	CCL	—	—	—	—	50.8	40.7	45.2	—	—	—	—	53.6	47.4	50.3
	UPP	—	—	—	—	52.8	37.2	43.7	—	—	—	—	60.0	46.6	52.4

Table 6.4: Performance on `wsj10` and `swbdnxt10` for models using words and POS tags only. Bold scores indicate the best performance of all models and baselines on that measure.
† Significantly different from best non-uniform baseline (italics) by a stratified shuffling test, $p < 0.01$; *: $p < 0.05$.

The left half of Table 6.4 presents results on `wsj10`. For the baseline models, the first column with horizontal text indicates the *input*, while for the backoff (Wds+POS) models, the first column with horizontal text indicates whether and how the extra

stream is modeled in dependents (as described in Section 6.2.3). The EM model with POS input is largely a replication of the original DMV, differing in the use of separate train, dev, and test sets, and possibly the details of the harmonic initializer. Our replication achieves an undirected attachment score of 63.8 on the test set, similar to the score of 64.5 reported by Klein and Manning (2004) when training and evaluating on all of `wsj10`. Cohen and Smith (2008) use the same train/dev/test partition that we do, and report a directed attachment score of 45.8, similar to our directed attachment score of 46.4.

The VB model which learns from POS tags does not outperform the EM model which learns from POS tags, suggesting that data sparsity does not hurt the DMV when using POS tags. As expected, the words-only models perform much worse than both the POS input models and the uniform LH baseline. VB does improve the words-only constituency performance.

The Cond. and Indep. backoff models outperform the POS-only baseline on all measures, but the Joint backoff model does not demonstrate a clear advantage over the POS-only baseline on any measure. The success of the Indep. model indicates that modelling dependent word identity does provide enough information to justify the increase in sparsity. The failure of the Joint model to provide a further improvement indicates that the extra information in the full joint over dependents does not justify the large increase in parameters. We also see that several models outperform the LH baseline on dependencies, but the advantage is much less in F-Score, underscoring the loss of information in the conversion of dependencies to constituencies. Finally, all models outperform CCL and UPP on F-score, emphasizing their reliance on the punctuation we removed.

6.3.6 Results: `swbdnxt`

The right half of Table 6.4 presents performance figures on `swbdnxt10` for input involving words and POS tags. As expected, the EM and VB baselines perform best when learning from gold-standard POS tags, and we again see no benefit for the VB POS-only model compared to the EM POS-only model. The POS-only baselines far outperform the uniform-attachment baselines on the dependency measures; to our knowledge this is the first demonstration outside the newspaper domain that the DMV outperforms a uniform branching strategy on these measures.

The other comparisons among systems listed in Table 6.4 are largely inconclusive.

			Dependency			Constituency		
		UNK	Dir.	Undir.	NED	P	R	F
EM	Wds	25	30.6	50.9	66.8	45.4	47.1	46.3
	Wds×Dur	25	26.1	46.5	62.0	45.6	48.7	47.1
	Wds×BI	25	23.1	44.6	59.5	45.4	47.5	46.4
VB	Wds	25	*36.4*	*55.1*	*73.0*	49.1	50.0	49.6
	Wds×Dur	25	31.8	51.7	71.3	49.2	*55.9*	*52.3*
	Wds×BI	25	23.4	45.5	62.5	*49.7*	53.2	51.4
BI+Wds	Cond.	25	30.9†	54.6	75.1†	59.3†	74.2†	65.9†
	Joint	50	31.1†	54.2	74.1	59.9†	72.8†	65.7†
	Indep.	25	34.6†	55.2†	73.2†	54.9†	59.3	57.0†
Dur+Wds	Cond.	25	32.6†	55.1	74.5†	59.1†	71.4†	64.7†
	Joint	50	31.8†	51.8†	70.8*	54.4†	60.5†	57.3†
	Indep.	50	**40.3**†	**59.1**†	**76.0**†	56.1†	61.7†	58.8†
	LH	—	24.1	50.8	72.7	**60.8**	**82.5**	**70.0**
	RH	—	29.2	52.0	57.9	22.2	30.1	25.5
	CCL	—	—	—	—	53.6	47.4	50.3
	UPP	—	—	—	—	60.0	46.6	52.4

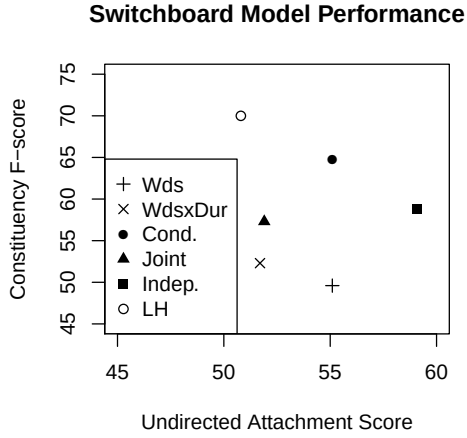


Table 6.5: Performance on `swbdnxt10` for models using words and duration. The scatterplot includes a subset of the information in the table: F-score and undirected attachment accuracy for backoff models and VB and LH baseline.
Bold, italics, and significance annotations as in Table 6.4.

Models do comparatively well on *either* the constituency or dependency evaluation, but not both. The backoff models outperform the baseline POS-only models in the constituency evaluation, but underperform or match those same models in the dependency

evaluation. Conversely, most models outperform the LH baseline in the dependency evaluation, but not in the constituency evaluation. There are probably two causes for the ambiguity in these results. First, the noise in the dependency gold-standard may have overwhelmed any advantage from backoff. Second, as we saw with `wsj10`, the conversion from dependencies to constituencies removes information, which may explain the failure of any model to outperform the LH baseline in the constituency evaluation.

Table 6.5 presents performance figures on `swbdnxt10` for input involving words and duration, including a scatter-plot of Undirected attachment against constituency F-Score for the interesting comparisons. In the scatter-plot, models up and to the right performed better, and we see that the negative correlation between the dependency and constituency evaluations persists in words and duration input. VB substantially outperforms EM in the baselines, indicating that good smoothing is helpful when learning from words. Other comparisons are again ambiguous; the dependency evaluation is noisy, and backoff models outperform baseline models on the constituency evaluation but not the LH baseline. Still, the backoff models outperform all words-only baselines in constituency score, with two performing slightly worse in dependency score and the Dur+Wds Indep. model performing much better. Among backoff models, we also don't see a systematic advantage of word duration over break index, or vice versa. For example, the Dur+Wds Indep. model outperforms the BI+Wds models in dependency score, but underperforms them in constituency score. So there is some evidence that word duration is useful, but we will find clearer evidence on the `Large Brent` corpus.

6.3.7 Results: `Large Brent`

Table 6.6 presents results on the `Large Brent` dataset. VB is even more effective than in the other datasets for improving performance among baseline models, leading to double-digit improvements on some measures. Moreover, the best dev-set UNK cutoff drops to 1 for all VB models, indicating that, on this dataset, VB provides good smoothing even in models without backoff. This difference between datasets is likely related to differences in vocabulary diversity; the type:token ratio in the `Large Brent` training set is about 1:15, compared to 1:5 and 1:9 in the `wsj10` and `swbdnxt10` training sets, respectively.

More importantly for our main hypothesis, all three backoff models using words and duration outperform the words-only baselines (including CCL and UPP) on all

			Dependency			Constituency		
		UNK	Dir.	Undir.	NED	P	R	F
EM	Wds	25	36.9	56.3	70.7	52.4	69.5	59.8
	Wds×Dur	25	31.3	51.1	66.9	50.7	64.7	56.9
VB	Wds	1	*51.2*	*64.2*	*77.3*	63.3	*68.1*	*66.0*
	Wds×Dur	1	47.0	60.5	74.0	***66.2***	64.9	65.5
Dur+Wds	Cond.	1	53.1*	65.5*	78.7*	65.4	68.6	**67.0***
	Joint	1	50.7	63.0	76.3	65.6	65.4†	65.5
	Indep.	1	**53.2**	**66.7**†	**79.6**†	61.5†	67.9	64.5
	LH	—	28.3	53.6	78.3	47.9	**85.6**	61.4
	RH	—	27.2	48.8	61.1	26.2	46.8	33.6
	CCL	—	—	—	—	41.7	58.8	48.8
	UPP	—	—	—	—	56.8	63.8	60.1

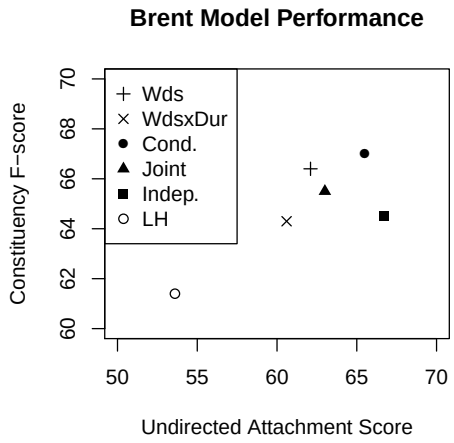


Table 6.6: Performance on `Large Brent` for models using words and duration. The scatterplot includes a subset of the information in the table: F-score and undirected attachment accuracy for backoff models and VB and LH baseline.
Bold, italics, and significance annotations as in Table 6.4.

dependency measures—the most accurate measures on this corpus, which has hand-annotated dependencies—and the Cond. model also wins on F-score.

6.3.8 Discussion

These experiments indicate that word duration and prosodic structure are useful for learning about syntax, in both adult-directed and child-directed speech. As a secondary result, these experiments show how fully-lexicalized models, with neither punctuation

nor gold-standard POS tags in the input, can perform well if they are smoothed properly.

Unlike Chapter 5, we did not find a systematic advantage of word duration cues over break index cues. This difference may be due to differences in the expressive power of the models: all the models of Chapter 5 could express only linear dependencies, while this chapter explored models of hierarchical structure. If break index encodes only relatively non-local information about syntax, a first-order HMM chunker would probably not be able to reveal this information, but a fully-hierarchical grammar could. By the same token, since word duration was useful in both set-ups, it remains possible that word duration conveys the non-local information of break index, although more noisily, and the local information exploited by the HMM chunkers.

In the next section, to develop a more qualitative understanding of the relative role of break index and word duration in our models, we embark on an investigation of what kind of information the models learned about syntax from word duration and break index. Specifically, we look at how the probability distributions over generative steps change before and after considering word duration and break index, characterizing the information about syntax encoded by word duration and break index.

6.4 Predictability or Prosodic Bootstrapping?

As was pointed out in Chapter 5, showing that acoustic cues are useful for unsupervised parsing does not itself clarify whether those cues are useful by way of syntactic predictability or by way of prosodic structure. In this section, we take a look at the posterior probability distribution over trees estimated by these models to characterize what kinds of information they exploited.

As a first step towards understanding how word duration relates to prosodic structure, Figure 6.7 presents the distribution over break indices for each duration tercile in the `swbdnxt10` dataset. The correspondences are not perfect, but we do see the general patterns that would be expected. First, a plurality of Long heads have a break index of 4, while only a minority of Short and Medium heads do, indicating that slow pronunciations do provide some evidence of an intonational phrase break. Similarly, remember that a break index of 2 indicates unusual phrasal phenomena, such as an intermediate phrase break that is missing its boundary tone. If we consider break indices of 2 and 3 together, Long pronunciations have a clearly stronger association with these break indices than do Medium or Short, providing further evidence of an association between

Long pronunciations and prosodic breaks. Conversely, over half of Short and Medium heads have a break index of 1 or 0, but only a quarter of Long pronunciations do, showing that there is an association between short pronunciations and weak prosodic breaks.

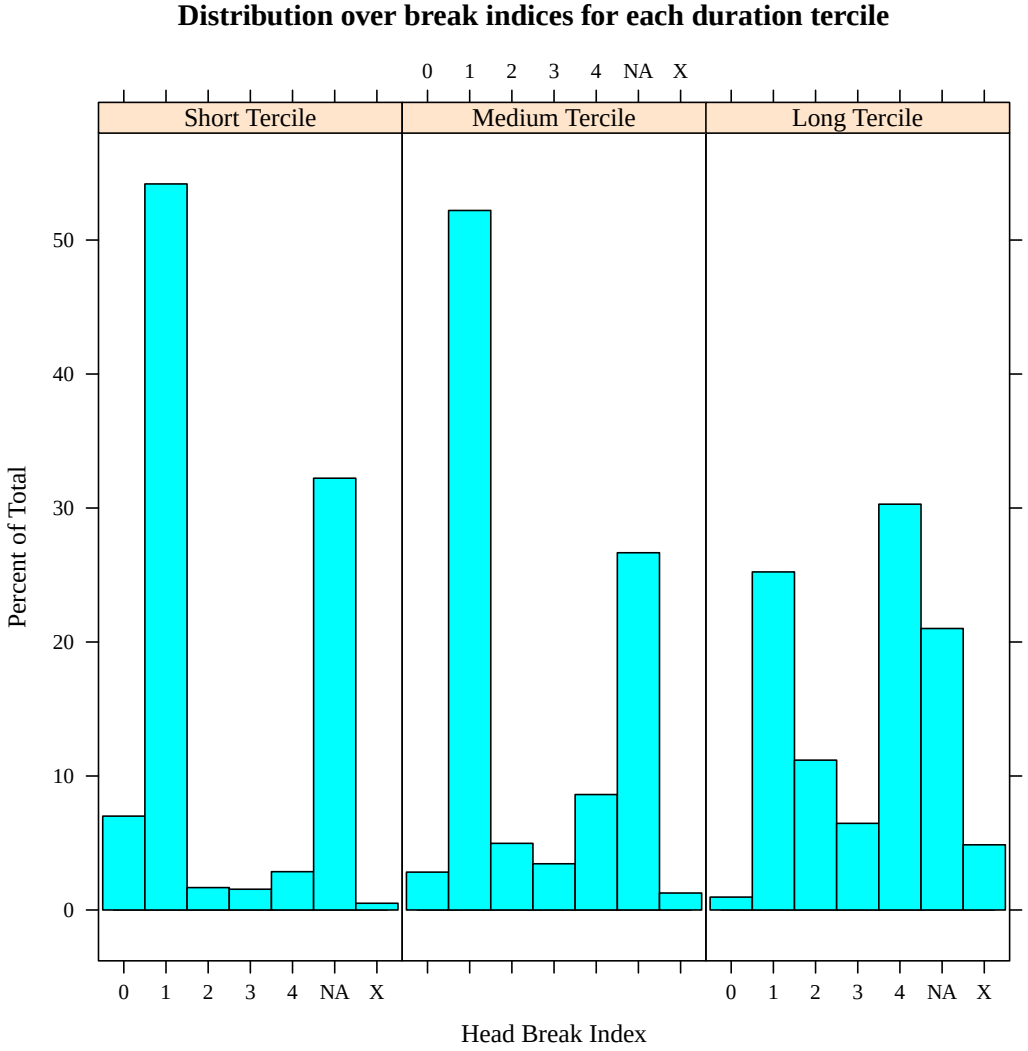


Figure 6.7: Distribution over break indices for each duration tercile on `swbdnxt10`.

So it is possible that this representation of word duration provides useful, albeit noisy, information about prosodic phrasing, and more work must be done to determine the roles of prosodic structure and syntactic predictability in the success of our models. We will do this by first noting that Prosodic Bootstrapping and Predictability Bootstrapping make different predictions about what kinds of words should be informative. The next section makes these different predictions explicit, and introduces "generative entropy" as a measure of uncertainty over parse trees that is suitable for testing these

predictions. The following section examines the generative entropy of various models on `swbdnxt10` and `Large Brent` when learning from words, words and break index, and words and word duration.

6.4.0.1 Method

To assess the role of prosodic structure and syntactic predictability in our models, we will exploit the fact that Prosodic Bootstrapping and Predictability Bootstrapping rely on different kinds of durational information in two different ways.

First, under Prosodic Bootstrapping, we would expect word duration to be useful by cuing phrasal boundaries: especially long words occur at the ends of phrases, and especially short words occur within a phrase. Concretely, this means that Long and Short words, or words with the strongest and weakest Break Indices of 4 and 0, should be unambiguous about the Dependents and Boundaries they generate, while Medium words should be more ambiguous. Under Predictability Bootstrapping, however, we would expect Short words to be informative because they pick out a single high-probability structure, while Long words should be ambiguous by ruling out the single high-probability structure but leaving many low-probability structures.

Second, Predictability Bootstrapping relies on a correlation between syntactic probabilities and word duration. Even a model that learns only from words is estimating syntactic probabilities, and, if Predictability Bootstrapping is feasible, should end up estimating more unambiguous probability distributions for Short words, even though the model doesn't know those words are Short. Thus, under Predictability Bootstrapping, we expect the posterior under a Words-Only model to exhibit the right correlation between word duration and syntactic probability, and for this correlation to simply be stronger under a Words and Word Duration Model. Prosodic Bootstrapping, however, relies on a statistical dependency between syntactic structure and prosodic structure. Models that learn from only words are not inducing prosodic structure, at least not in any direct and systematic way, so we would not expect the posterior under a Words-Only model to show the right correlation.

To make these predictions quantifiable, we will introduce and examine the "generative entropy" of each model over a training corpus when learning from words, words and break index, or words and word duration. As introduced in Section 6.2.1, at each step of the generative story of a parse tree, a given head word has several choices: generate a boundary, or generate another dependent in the current direction. The "generative entropy" of a specific step in the generation of a parse tree (and sentence) is just

the entropy of the probability distribution over these possible decisions.

More specifically, one step in a generative story is defined by a triple specifying the head word under consideration, the direction it is currently generating, and the extent of the dependents the head word already has in that direction. Concretely, if we are considering the (0-based) t^{th} word in the string as our head word h, generating to the right, with no dependents yet, then we are considering the probability distribution over what h can generate to the right when its rightward dependents span $(t+1,t+1)$ i.e. it has none. For another example generative step, h has dependents that span 2 words, and we are consider the probability distribution over rightward generations when h has rightward dependent subtrees that span $(t+1,t+3)$. Note that we do not care *how* the dependents are organized: h may have both words as immediate dependents, or it may have only one word as an immediate dependent which in turn takes the other as a dependent. This means that we are marginalizing over previous decisions in the generative story.

These probability distributions are over generating a Bound and generating each of the words beyond the current set of dependents as a Dependent. For example, if we are considering the first word h that has dependents spanning the next two words in a sentence 5 words long, at this step h could either generate a Boundary, or it could generate one of the last two dependents. The entropy of this probability distribution then tells us how uncertain we are for this step. We will examine the generative entropy of every generative step of every derivation of every sentence (the relevant probability distributions can be efficiently computed from the packed chart that results from the Inside-Outside algorithm).[5] Once we have these entropies, we will see how they vary according to head word duration and head word break index.

Next, we will look at generative entropies for models that learn from Words and Break Index. If the generative entropies match the two predictions made above about Prosodic Bootstrapping, then we have some evidence that prosodic phrase structure does co-vary with syntax in the kind of way predicted by the Prosodic Bootstrapping hypothesis. We will then continue to look at the generative entropies for models that learn from Words and Word Duration, and see if they match the predictions above for Predictability Bootstrapping, Prosodic Bootstrapping, or neither.

[5] We do not consider the generation of sentence roots, as these have no break indices or durations.

6.4.0.2 Words and Break Index

As just discussed, there are two hallmarks of Prosodic Bootstrapping we want to look for. First, the posterior under a Words-Only model should look very different from a posterior that is informed by prosodic structure. Second, when learning from break index, head words with break indices of 0 and 4 should have the lowest generative entropy.

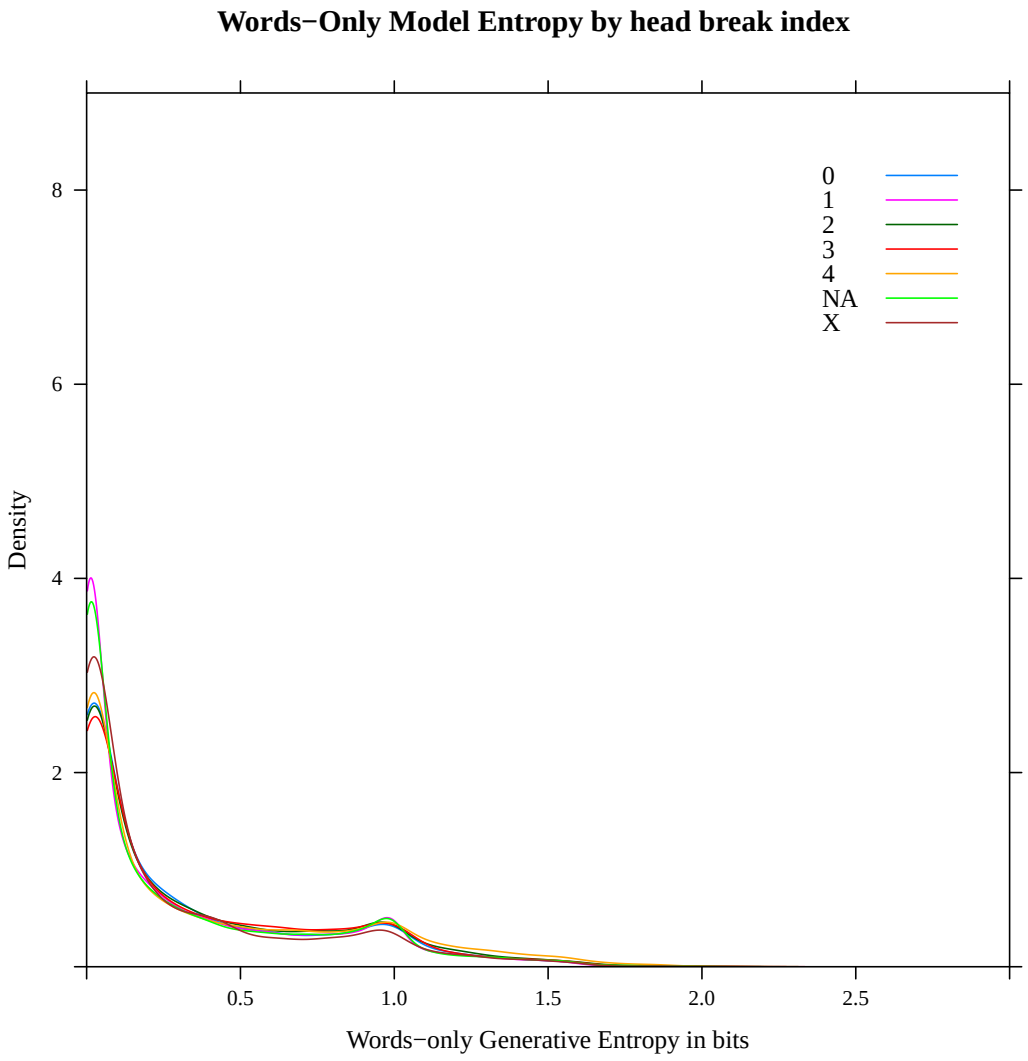


Figure 6.8: Distribution over words-only generative entropy for each break index on `swbdnxt10`.

We first look at generative entropies on the `swbdnxt10` corpus from the Words-Only VB baseline with an UNK cutoff of 25 (the same model evaluated in Table 6.4). Figure 6.8 presents a kernel density plot, a kind of smoothed histogram, over generative

entropies from this model by break index. The horizontal axis is the entropy of a step in the generative story, and the height of the colored line increases as more steps have a generative entropy in that region. Ignoring distinctions between the different break indices for the moment, we see that the distribution of generative entropies is broadly bimodal. There is a large peak near zero, a dip, and a small rise near 1. The generative entropies are in bits, so the left-most peak indicates a concentration of steps which are almost completely unambiguous: there are many steps in which one generative possibility gets almost all of the mass. Additionally, almost all generative steps have an entropy between 0 and about 1, indicating that most generative steps involve no more than (effectively) one yes/no decision. Note that this doesn't mean the posteriors over trees are this unambiguous, since it is often not possible to construct a full projective dependency tree from only the high-probability generative steps.

Looking to differences between the different break indices, we see that most break indices have generally the same profile. Break indices of 0 and "NA" appear to have the most very low-entropy steps, with "X" not far behind. A word was annotated with break index "NA" if it was outside of the fluent prosodic phrases selected for ToBI annotation, and "X" is reserved for words with unclear prosodic phrasing. The low ambiguity of these break indices suggests that the words-only model has learned something about disfluencies, perhaps because of the presence of filler words in the input such as "umm." The low ambiguity of break index 1 suggests that phrase-internal syntax may be overall more regular than syntax across phrases. Overall, as predicted by Prosodic Bootstrapping, we do not see a clear influence of prosodic phrase boundaries on generative entropy.

Next, we will examine the generative entropy of the Independent emissions model when learning from Words and Break Index on `swbdnxt10`. We calculate generative entropy on the training set with an UNK cutoff of 25 (the same model evaluated in Table 6.4). Figure 6.9 presents a kernel density plot over generative entropies as computed by this model by break index. We see that the shape of the density plot is broadly the same as under the Words-Only model, but the details by break index are different. Most notably, break indices of 4 now very frequently have low generative entropy. This is consistent with the Prosodic Bootstrapping prediction that the placement of phrase boundaries should be informative about syntactic events.

Figure 6.10 presents a more detailed version of the distribution over generative entropies, dividing out generative steps according to their direction and valence.[6] For

[6]Break index "X" has been excluded, as it virtually always gives almost all probability mass to

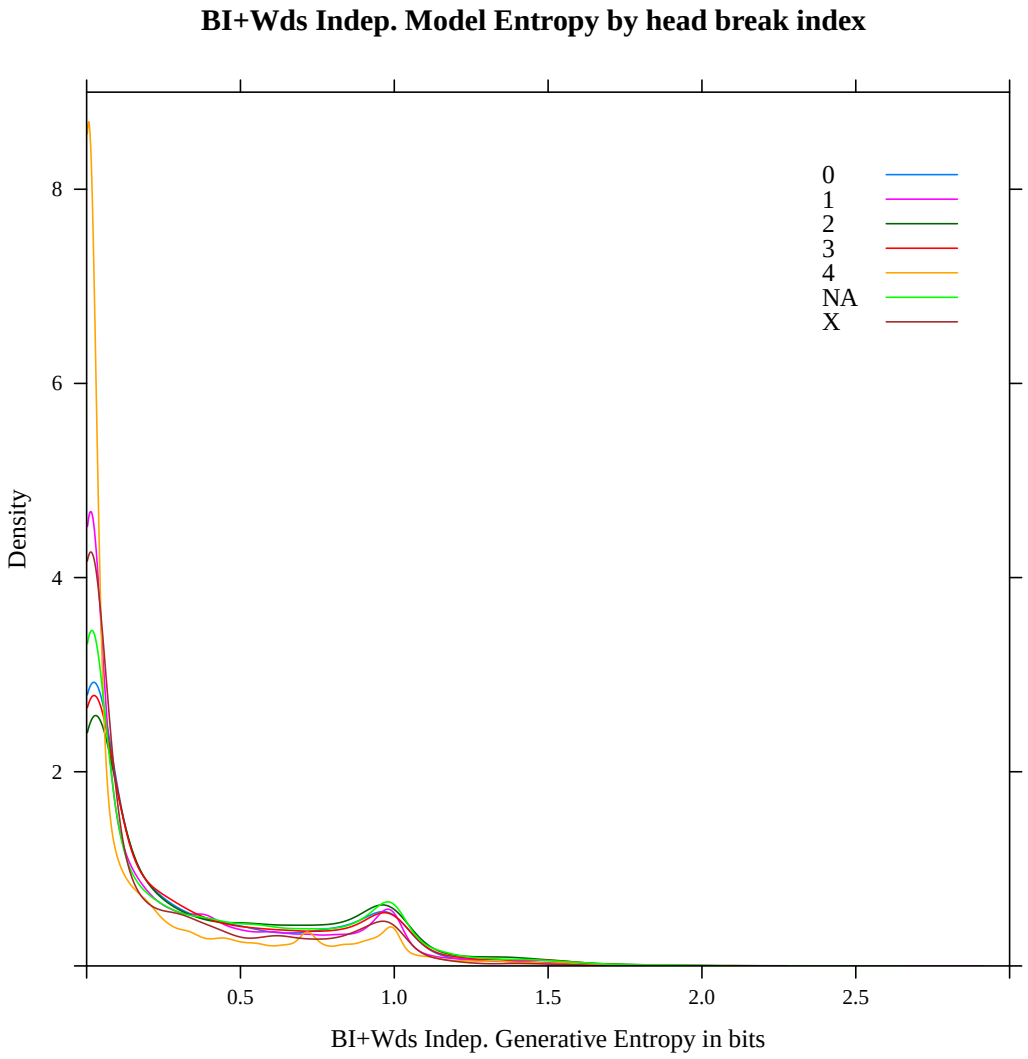


Figure 6.9: Distribution over BI+Wds Indep. Model generative entropy for each break index on `swbdnxt10`.

leftward generative steps, we see a clear influence of prosodic breaks, as break index 4 exhibits the most very low entropy steps, and break index 3 exhibits the second most very low entropy steps. For rightward generative steps before taking any dependents, we see a clear influence of prosodic phrase continuity, as break index 0 exhibits the most very low entropy steps, and break index 1 exhibits the second most. In broad outline, this is consistent with the predictions of Prosodic Bootstrapping: the most extreme prosodic phrase breaks, 0 and 4, appear to convey different kinds of information

generating a Boundary before taking any dependents. This led to plots that were unreadable when the vertical axis scale was adjusted to accept the glut of extremely low-entropy "X" heads that never take any dependents.

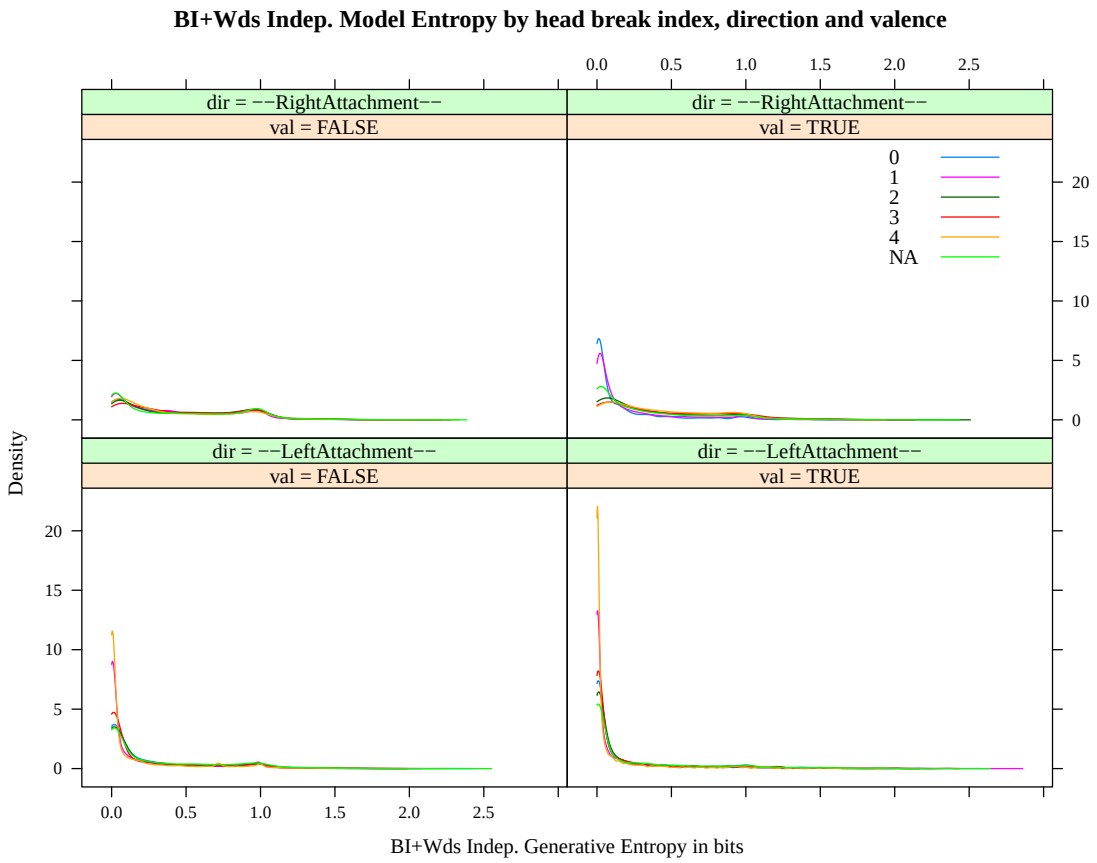


Figure 6.10: Distribution over BI+Wds Indep. Model generative entropy for each break index on `swbdnxt10`, split up by step direction and valence.

from each other, while conveying the same kind of information as their less extreme counterparts, 1 and 3, with more certainty.

However, the details of these results suggest that the Prosodic Bootstrapping hypothesis should be revisited. If prosodic breaks cue syntactic breaks by coinciding with syntactic breaks, break indices of 3 and 4 should be informative for *rightward* events. Instead, they are informative for *leftward* events. Moreover, if we look at the actual probability distributions (not presented) rather than just the entropy, we find that these 3-and-4-headed leftward steps overwhelmingly prefer to generate a Boundary to the left. That is, these results seem to indicate that prosodic breaks cue syntactic breaks by happening *after* them. In any event, these results do indicate that prosodic phrase structure interacts with syntax in a way that can be learned from words and break index, which is consistent with the overall predictions of the Prosodic Bootstrapping hypothesis.

All together, these results suggest that the Words and Break Index models were

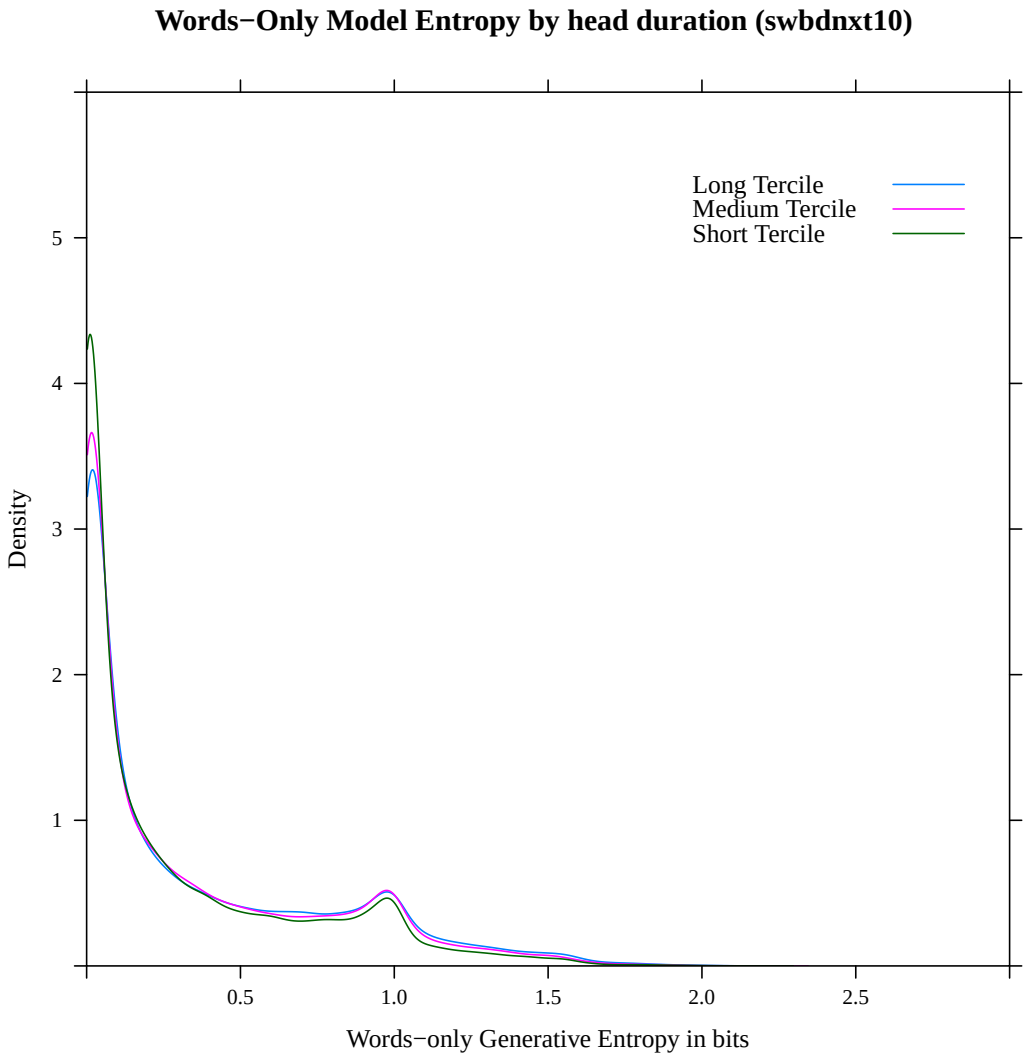


Figure 6.11: Distribution over words-only generative entropy for each duration tercile on `swbdnxt10`.

doing some kind of Prosodic Bootstrapping. They learned a correspondence between Break Index and syntax that appeared to reflect prosodic phrase structure, while the correspondence learned by the Words-Only models reflected only disfluencies. Next, we turn to the models that learned from Words and Word Duration.

6.4.0.3 Words and Word Duration

Under Predictability Bootstrapping, we expect two things about the generative entropies. First, Short pronunciations rule out the many low-probability possibilities, while Long pronunciations rule out the few high-probability possibilities, so we ex-

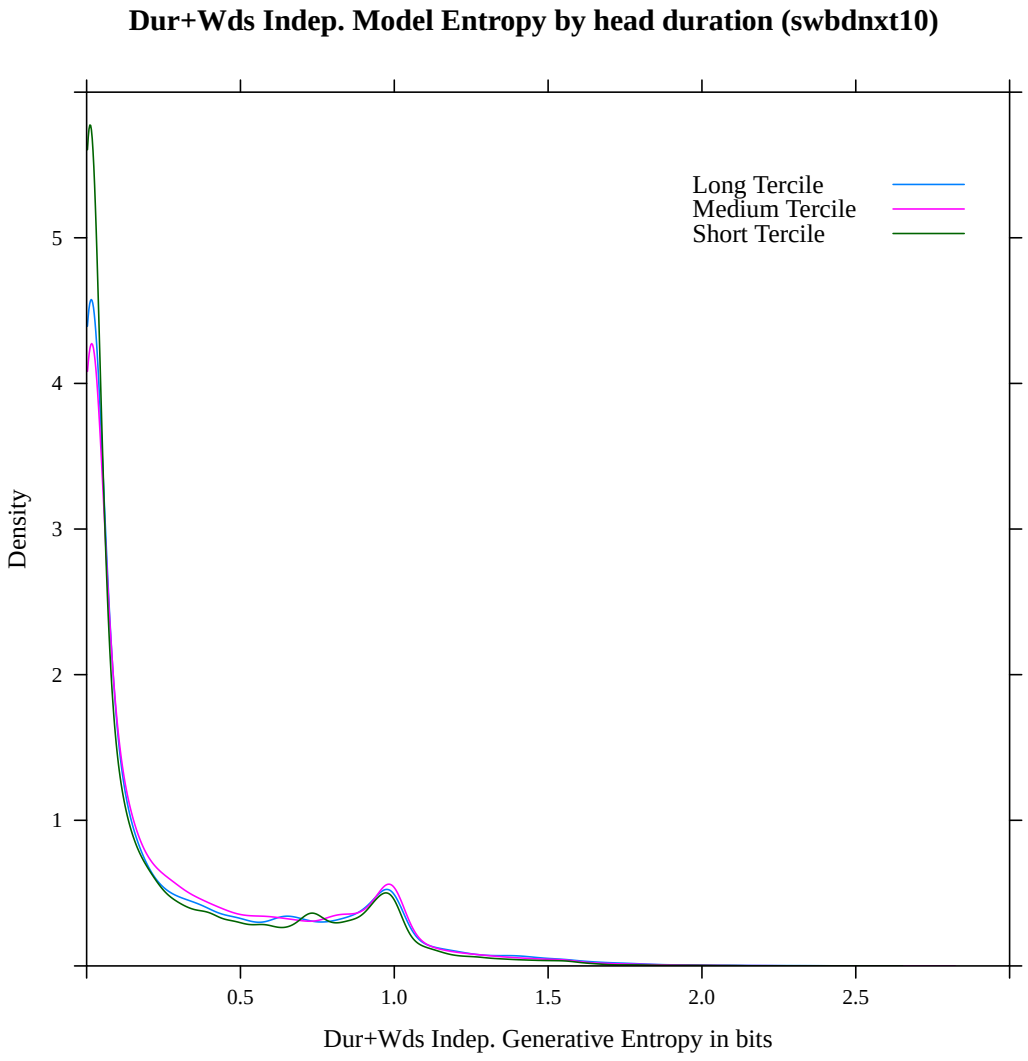


Figure 6.12: Distribution over Dur+Wds Indep. Model generative entropy for each duration tercile on `swbdnxt10`.

pect Short pronunciations to have generally lower entropy than Long pronunciations. Second, since Words-Only models are learning about syntactic probabilities, we expect the Words-Only posterior to exhibit a weaker version of the relevant correlation between word duration and generative entropy.

Again using the Words-Only VB baseline with an UNK cutoff of 25 for our Words-Only model, Figure 6.11 presents a kernel density plot over generative entropies from this model by duration tercile. The set of generative entropies is the same as in Figure 6.8, but Figure 6.11 divides them according to the duration of the head of each generative step rather than according to the break index, so the overall shape of the density plot is basically the same. However, we do see hints of the relevant correla-

tion between duration tercile and generative entropy: Short heads have low generative entropy more often than do Medium heads, which in turn have low generative entropy more often than Long heads do. Notably, this is the correlation expected under Predictability Bootstrapping, not Prosodic Bootstrapping. Moreover, as predicted by Predictability Bootstrapping, the relevant correlation shows up even in the Words-Only posterior.

Next, we examine the generative entropy of the Independent emissions model when learning from Words and Word Duration on `swbdnxt10`. We calculate the generative entropy on the training set, this time using an UNK cutoff of 50 (the same model evaluated in Table 6.4). Figure 6.12 presents a kernel density plot over generative entropies as computed by this model by duration tercile. This density plot is broadly similar to the one computed from the Words-Only model, but now there are many more steps with low generative entropy. There is arguably evidence for both Prosodic Bootstrapping and Predictability Bootstrapping in this posterior. Short heads more often have low generative entropy than Long heads, so the model exhibits the correlation underlying Predictability Bootstrapping. On the other hand, Long heads are no longer the most ambiguous, indicating that the Wds+Dur model learned something extra about long words.

Figure 6.13 presents a more detailed version of the distribution over generative entropies, parceling out generative steps according to their direction and valence. We see the correlation expected by Predictability Bootstrapping among generative steps before taking any dependents: Short heads are most often very low-entropy, and Long heads are least often very low-entropy. However, among leftward steps after already taking at least one dependent, Long heads are most often very low entropy. Intriguingly, we saw in the Wds+BI posterior that, for this type of step, heads of Break Index 3 and 4 were most often low-entropy. Thus, we see evidence of Predictability Bootstrapping among generative steps before taking any dependents, and tentative evidence of Prosodic Bootstrapping among leftward generative steps after taking at least one dependent.

Finally, we turn to the `Large Brent` data to see if word duration interacts with syntactic probabilities in the same way in child-directed speech. Figure 6.14 presents the generative entropy of the Words-Only model with an UNK cutoff of 1 on the `Large Brent` training set divided by head duration tercile. Encouragingly, we see the pattern expected under Predictability Bootstrapping: Short heads are most often low-entropy, while Long heads are least often low-entropy.

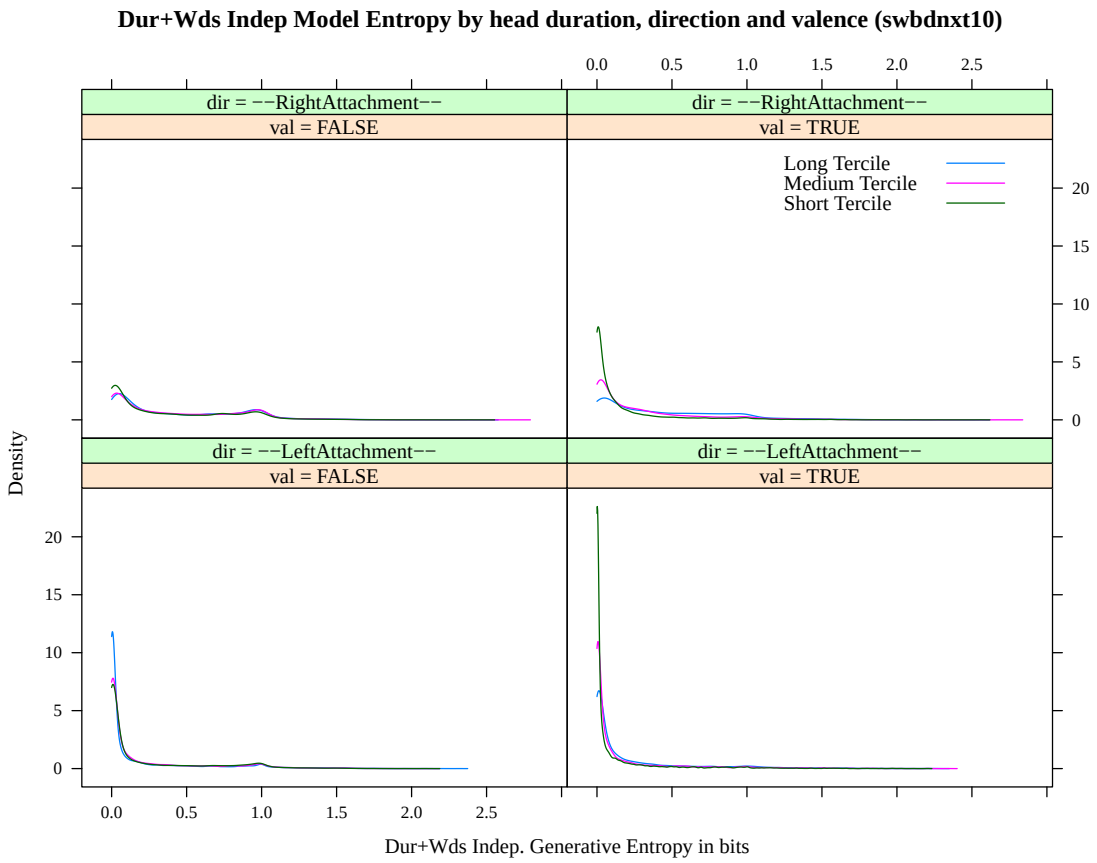


Figure 6.13: Distribution over Dur+Wds Indep. Model generative entropy for each duration tercile on `swbdnxt10`.

To see how the posterior changes when considering word duration, Figure 6.15 presents the generative entropy of the Words and Word Duration Independent emissions model according to word duration. We see a dramatic rise in the number of very low entropy Short heads, and a moderate rise in the number of very low entropy Medium and Long heads. Together, this means that, under the Words and Word Duration Model, the correlation between head duration and head entropy is even stronger, as expected under a model engaged in Predictability Bootstrapping.

Figure 6.16 presents the generative entropy of the Words and Word Duration Independent emissions model according to head duration and generative direction and valence. We see the correlation expected by Predictability Bootstrapping in generative steps before taking any dependents in both directions, with virtually all Short pronunciations having very low entropy. After taking dependents, generative steps become much more ambiguous for heads of all lengths. Among rightward generative steps after taking one or more dependents, we see that Short heads have low entropy most

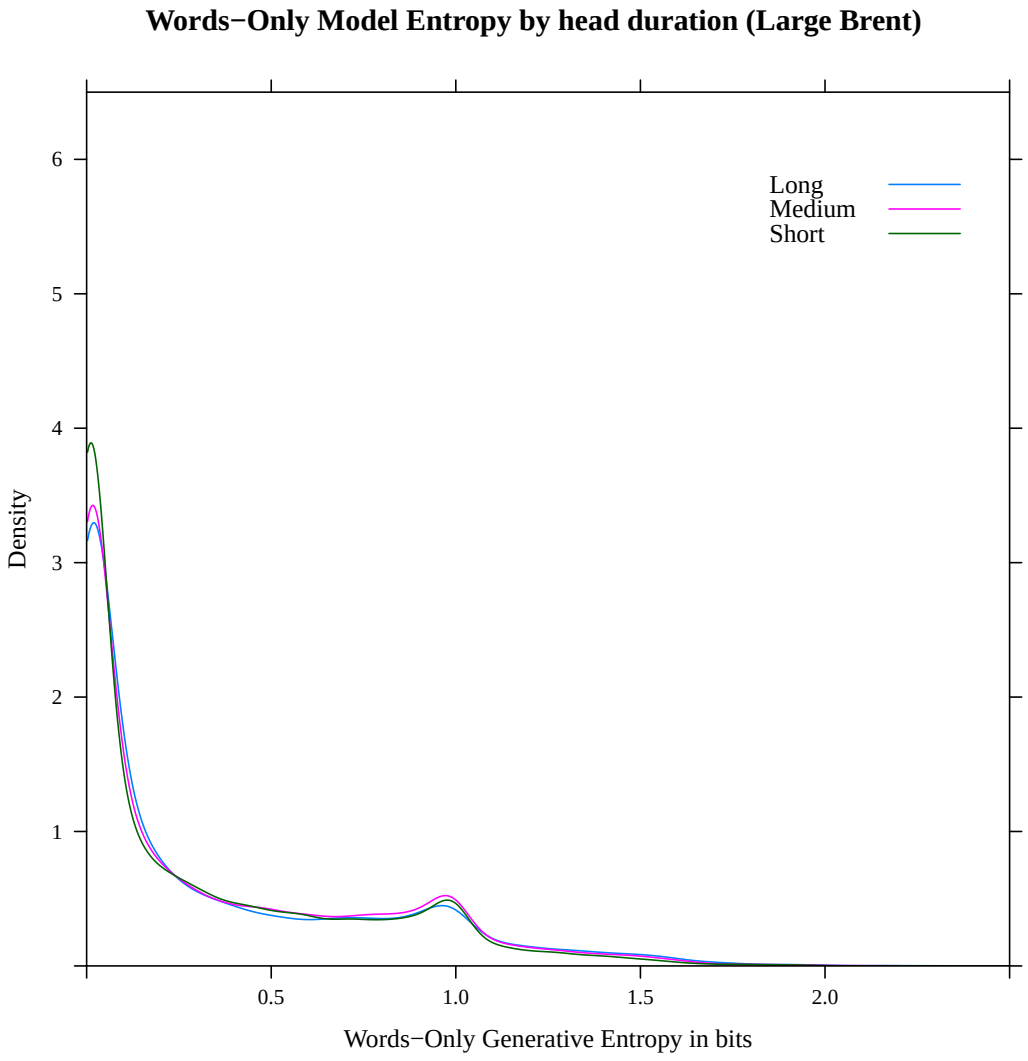


Figure 6.14: Distribution over Words-Only Model generative entropy for each duration tercile on `Large Brent`.

often, but the correlation has weakened substantially compared to before taking any dependents.

At the scale of these axes, there's no evident correlation between head duration and generative entropy among non-first leftward steps. If we zoom in to a more appropriate scale (not pictured), there is a slight correlation, but in the wrong direction: Long heads are most often low entropy, and Short heads are least often low entropy. Although this correlation is weak, we saw the same trend in the `swbdnxt10` data, both for Long heads in non-first leftward steps, and heads of break index 4 in non-first leftward steps. Accordingly, we see very strong evidence of Predictability Bootstrapping among first generative steps, moderate evidence of Predictability Bootstrapping among subsequent

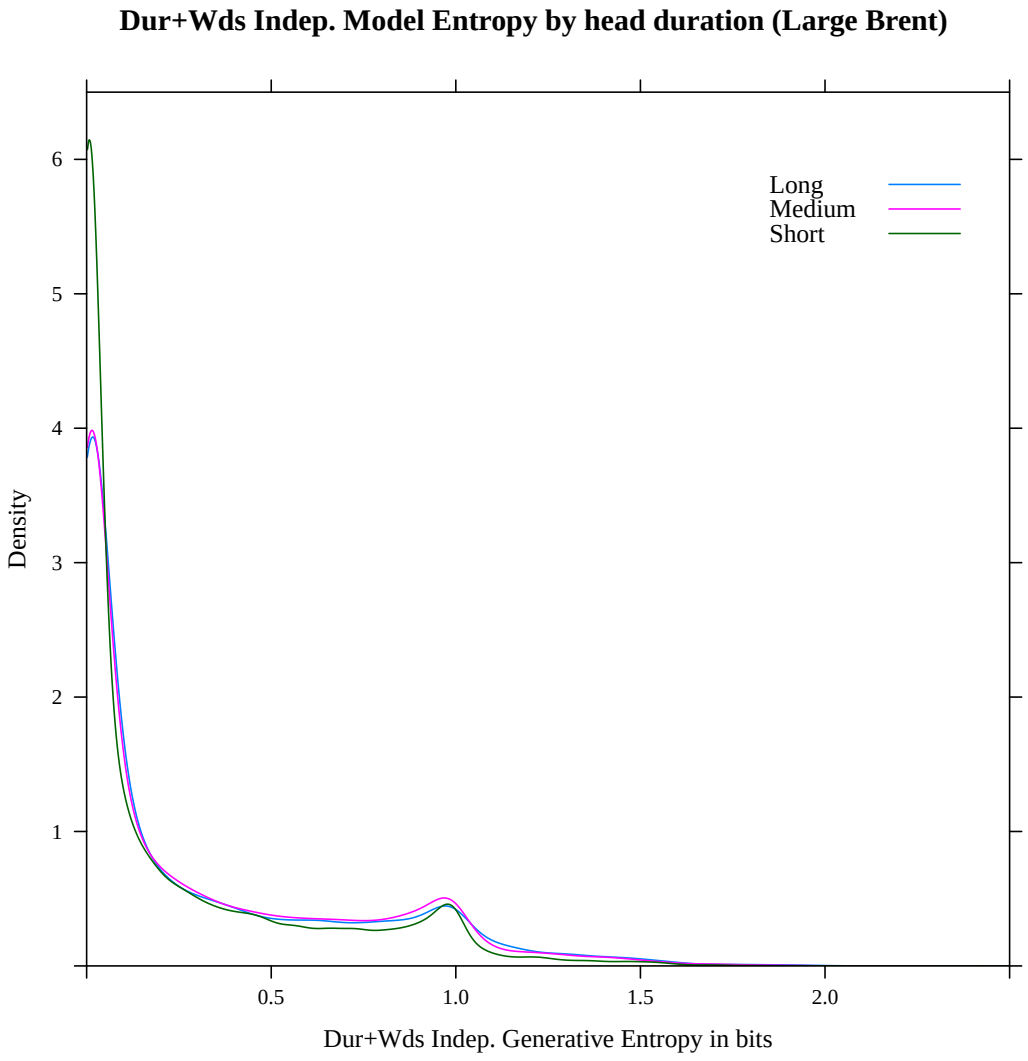


Figure 6.15: Distribution over Dur+Wds Indep. Model generative entropy for each duration tercile on `Large Brent`.

rightward steps, and perhaps some evidence of at least the beginnings of Prosodic Bootstrapping among subsequent leftward steps.

6.4.1 Conclusion

In this section, we set out to understand how different kinds of information influenced our models' certainty in building dependency trees. In particular, we wanted to see evidence of signatures of Prosodic Bootstrapping and Predictability Bootstrapping. Specifically, under Prosodic Bootstrapping, break index information should interact with syntax by way of prosodic phrase structure, and this interaction should be appar-

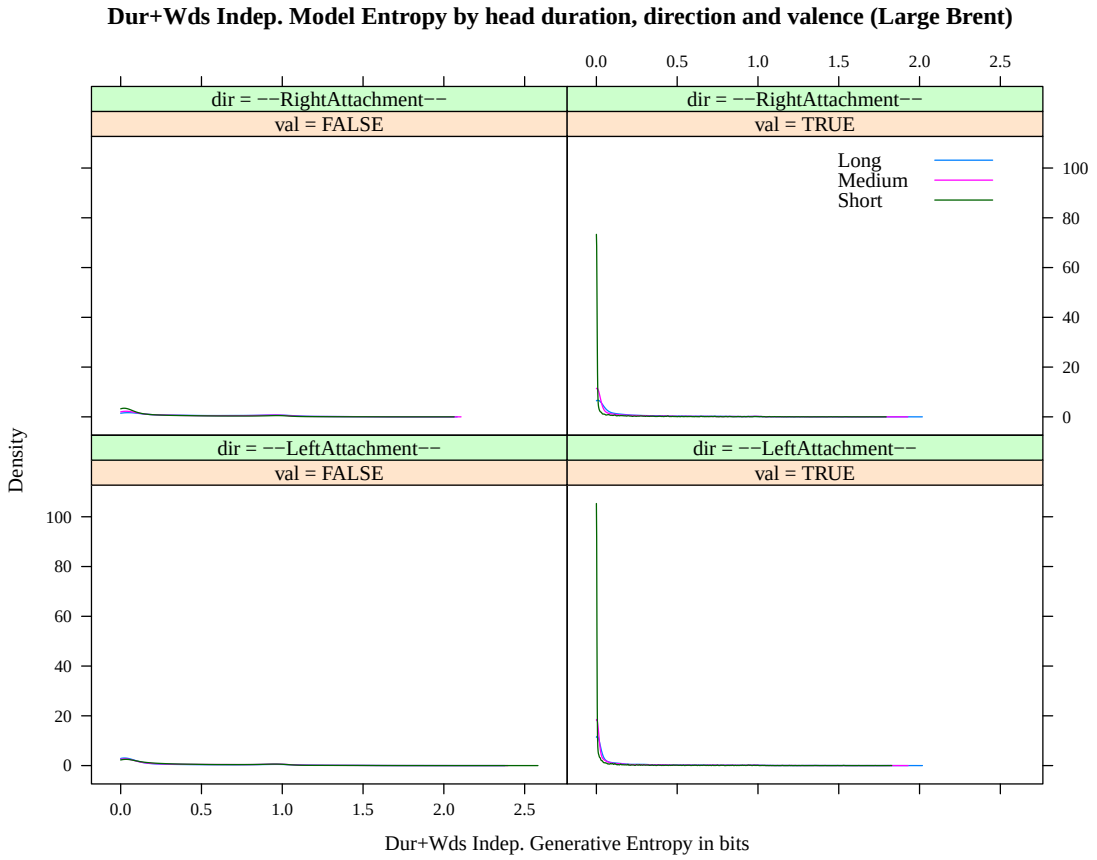


Figure 6.16: Distribution over Dur+Wds Indep. Model generative entropy for each duration tercile on `Large Brent`.

ent only when learning from both Words and Break Index, not when learning from Words alone. Under Predictability Bootstrapping, Short words should be less ambiguous about syntax than Long words in both the Words-Only and the Words and Word Duration models.

We found that, when presented with Words and Break Index information, the model behaved as predicted by Prosodic Bootstrapping, finding a correlation between phrase-final and phrase-internal break indices (as opposed to break indices associated with disfluent or unusual prosody) and generative entropies that was not evident in the Words-Only model. The details of this interaction, however, were not consistent with previous accounts of Prosodic Bootstrapping. These previous accounts predicted that prosodic breaks should cue syntactic breaks by coinciding with them, but our model learned to predict syntactic breaks that preceded prosodic breaks. On reflection, this is not actually that surprising because, as discussed in Section 3.2.3, prosodic trees and syntactic trees have systematically different shapes: prosodic trees are shallow and rel-

atively balanced but syntactic trees are deep and often unbalanced. It may be easy to align syntactic breaks with prosodic breaks once both structures are known, but it is not at all clear that this kind of alignment should be easy during learning.

We also found that, on both adult-directed and child-directed speech, the model behaved as predicted by Predictability Bootstrapping when presented with Words and Word Duration, with some evidence of Prosodic Bootstrapping for non-first leftward generative steps. The Words-Only posterior revealed that Short heads had the most low-entropy decisions to make and Long heads had the fewest, and this correlation strengthened when learning from Words and Word Duration. This investigation indicates that it is statistically feasible to bootstrap syntactic dependencies from word duration by way of both syntactic predictability and prosodic phrase structure, and that it is markedly easier to do so by way of syntactic predictability.

6.5 Discussion

The primary conclusion from these experiments is that word duration can provide an advantage in parsing performance, at least when training and evaluating on child-directed Speech, and that this advantage is probably mostly due to syntactic predictability, with a smaller contribution from prosodic structure. Experiments also suggest that the word type/token ratio in CDS is much lower than we find in newspaper text, and is in fact low enough in CDS (but *not* in newspaper text) to support a fully-lexicalized modeling approach with very low UNK cutoffs, as long as the model implements good smoothing.

The experiments also constitute a broad-coverage complement to the previous work of Gahl and Garnsey (2004), Gahl et al. (2006), and Tily et al. (2009). Gahl and Garnsey (2004) and Gahl et al. (2006) showed, in laboratory studies, that talkers reduced verbs and verb dependents when they were in the verb's preferred frame for a small set of carefully controlled materials containing dative alternations and transitive/intransitive frames. Tily et al. (2009) showed essentially the same thing in a corpus study (using the same base corpus as `swbdnxt10`) for the case of the dative alternation. In investigating the posterior generative entropy, we found that Short pronunciations corresponded to very low entropy in adult-directed and child-directed speech, suggesting that these kinds of syntactic predictability effects are pervasive throughout syntactic phenomena in spontaneous speech.

Chapter 7

Conclusion

This chapter summarizes the dissertation, addresses shortcomings, and describes future directions. Section 7.1 enumerates the contributions of this dissertation, Section 7.2 summarizes the work in support of each contribution in greater detail, Sections 7.3 describes future modeling directions, and Section 7.4 describes potential experiments to follow up on this dissertation.

7.1 Summary of Contributions

The primary contribution of this dissertation is the Predictability Bootstrapping hypothesis: the proposal that predictability effects may make language easier for infants to learn by providing them with observable clues, in the form of linguistic reduction, to the unobserved statistical knowledge of an adult speaker. In support of this hypothesis, this dissertation made one theoretical contribution and two empirical contributions.

For the theoretical contribution, Chapter 3 argued for a new view of bootstrapping accounts that is based on statistical dependencies. Specifically, we argued that a bootstrapping account is one that highlights a statistical dependency between different kinds of linguistic knowledge, and proposes that this statistical dependency facilitates language acquisition by reducing the space of possible grammars a child must consider or explore. The primary consideration in formulating a bootstrapping account, then, is the functional form of the dependency proposed: what kinds of variables does it relate, and how complex is this relation? This theoretical contribution provided *a priori* support for the Predictability Bootstrapping hypothesis because it relies on a very simple statistical dependency that should be easy to find and exploit.

It would be difficult for children to take advantage of predictability effects if they

were not present in child-directed speech. The corpus studies of Chapter 4 found that child-directed speech does exhibit at least some of the same predictability effects as does adult-directed speech, providing the first empirical contribution toward the Predictability Bootstrapping Hypothesis.

It would also be difficult for children to take advantage of predictability effects if influences of predictability on redundancy were weak, swamped by other factors, or more complex than anticipated. In the second empirical contribution of this dissertation, unsupervised statistical models were used to explore the feasibility of a specific version of the Predictability Bootstrapping hypothesis: that phonetic reduction in terms of word duration is useful for learning about syntax. Chapter 5 found that predictability effects in terms of word duration may be useful for learning a kind of shallow constituency syntax, and Chapter 6 found that predictability effects are useful for learning unlabeled dependency syntax.

Thus, by the theoretical contribution, predictability bootstrapping should be easy, by the first empirical contribution, predictability effects exist in the evidence children have, and by the second empirical contribution, effects of syntactic predictability on word duration are strong enough to be useful.

This dissertation also made two secondary empirical contributions. First, the corpus studies of Chapter 4 found that talkers modulate predictability effects according to listener and channel characteristics. This result directly relates predictability effects to aspects of the communicative scenario, providing a new kind of evidence that bolsters the case that predictability effects reflect a strategy towards more efficient communication. Second, we used the same computational models to evaluate the Prosodic Bootstrapping hypothesis: the proposal that prosodic structure provides children with useful information about syntax. As far as we are aware, these models constitute the first computational models of prosodic bootstrapping when the syntax is unobserved. Our models found that prosodic structure, as represented by ToBI, is useful for learning about syntax when the prosodic structure is known (i.e. the models observe hand-annotated break indices), but that it is much more difficult to exploit prosodic regularities when learning from durational reflexes of prosody.

There are two potential interpretations of this last contribution.

7.2 Summary of Work

This dissertation opened with the "Predictability Bootstrapping" hypothesis: the proposal that predictability effects may make language easier for infants to learn by providing them with observable clues, in the form of linguistic reduction, to the unobserved statistical knowledge of an adult speaker. Chapter 3 argued that bootstrapping accounts for language acquisition come down to the proposal that a statistical dependency between two kinds of linguistic knowledge reduces the dimensionality of the space a child must explore, and discussed how the effectiveness of this dimensionality reduction is sensitive to both the functional form and strength of the statistical dependency. Predictability Bootstraping relies on dependencies that should be both simple and strong: high-probability structures are more reduced, and low-probability structures are less reduced. Moreover, if predictability effects really are the result of functional adaptations towards efficient speech, as discussed in Chapters 2 and 4, then the basic form of this dependency should be relatively stable across languages. Chapter 4 proceeded to show that some of the more straightforward predictability effects that had been found in adult-directed speech also exist in child-directed speech. Together, these arguments and results motivate Predictability Bootstrapping as plausible from a learnability perspective.

This dissertation also evaluated in detail one instantiation of the Predictability Bootstrapping hypothesis: that reduction in terms of word duration should be useful for learning about syntax. To do so, we also examined the Prosodic Bootstrapping hypothesis, which relies on a different proposed dependency between word duration and syntactic structure. The Prosodic Bootstrapping hypothesis had not received a computational treatment, and devising one highlighted previously-unappreciated computational difficulties. Specifically, prosodic structure is itself a complex latent variable that is difficult to induce. Moreover, prosodic structures tend to be shallow while syntactic structures tend to be deep, which complicates the prospect of learning a statistical dependency between syntactic and prosodic structures.

The unsupervised syntactic chunking experiments of Chapter 5 and the unsupervised dependency parsing experiments of Chapter 6 did not rule out prosodic phrasing as a potential source of information about syntax for language-learning infants, but they did indicate that effects of syntactic predictability on word duration provide a simpler signal that is easier to exploit. Specifically, in our chunking experiments, the chunker that maintained a separate "prosody" stream outperformed the baseline mod-

els when learning from hand-annotated break index, indicating that prosodic phrasing provides a benefit. However, the chunkers that learned from word duration cues performed even better, with and without the separate "prosody" stream. Together, these results indicated that gold-standard prosodic phrase information was useful for unsupervised chunking, but that word duration measures were more useful, and were useful in a way that did not require an intermediate representation.

The unsupervised dependency parsing experiments of Chapter 6 provided more direct evidence for predictability effects as a signal about syntax, together with a more sophisticated view of the potential utility of prosodic phrase structure in early syntax learning. Specifically, we again found that break index and word duration were useful for our unsupervised model of syntax, but we did not find an overall advantage for word duration over break index. A more qualitative study of the posterior distribution over generative decisions provided evidence for bootstrapping from both predictability effects and prosodic phrasing. On both adult-directed and child-directed speech, the models that learned from words alone were more certain about the syntactic structure associated with shorter heads, and this certainty was strengthened dramatically in the models that also saw word duration. This increased certainty for shorter heads indicated that the models reflected and exploited the inverse correlation between word duration and syntactic certainty due to syntactic predictability.

The posteriors also exhibited evidence for prosodic bootstrapping, although a different kind of prosodic bootstrapping than is usually proposed. Specifically, while our models that saw break index indicated the existence of a useful relationship between prosodic boundaries and syntactic boundaries, prosodic phrase-final boundaries did not coincide with syntactic phrase-final boundaries but rather followed syntactic phrase-initial boundaries. We found some evidence for broadly the same relationship in the models that saw word duration: while short heads were overall most informative, long heads were most informative for non-first leftward decisions. Together, these results indicate that word duration may provide infants with information about syntax by way of both prosodic structure and syntactic predictability, but that syntactic predictability is easier to exploit.

The experiments in this dissertation are limited in a number of ways. For example, the models knew nothing about parts of speech, morphology, or semantics, and ignored the fact that learning takes place over time. These shortcomings can be addressed through elaborations on the models, and we will discuss specific potential elaborations shortly. However, by far the greatest limitation of the work in this dissertation is its

consideration of English data only. While Chapter 3 presented some reasons to think that predictability effects should be useful cross-linguistically, a proper treatment demands actual data in other languages, in the form of spontaneous speech treebanks, that is ideally gathered from caregiver-child interactions.

In addition to the results themselves, the work in this dissertation joins the work of others (e.g. Goldwater et al., 2009; Kwiatkowski et al., 2012; Johnson et al., 2010) in making a secondary methodological point about the utility of computational models in language acquisition. The developmental linguistics community is broadly divided into lexicalist empiricists (e.g. Tomasello, 2000; Bannard et al., 2009; Freudenthal et al., 2007; Chang et al., 2006a), who propose that early learning is word-specific and emphasize domain-general statistical methods, and early-abstraction nativists (e.g. Fisher, 2002; Pinker, 1994; Chomsky, 1980; Wexler and Culicover, 1983; Baker, 2002), who propose that early learning focuses on learning rules that abstract over words and emphasize the child's goal of hierarchical adult-like structures. The models in this dissertation demonstrate that we can "have our cake and eat it too," because recent advances in machine learning and natural language processing have made it possible to measure how well the input identifies complex structured objects. The most successful approaches in unsupervised parsing in NLP have sought to incorporate both concrete and abstract elements, together with both domain-general statistical methods and prior domain-specific knowledge, and it is unlikely that children track only domain-general surface statistics or purely abstract language-specific rules.

Early work on syntax acquisition emphasized the importance of understanding the extent to which the stimulus identified grammatical structures as an important line of evidence about language acquisition (e.g. Gold, 1967). However, computational methods at the time were too limited to assess the stimulus in a statistical sense, and, in the absence of clear evidence either way, the community bifurcated into those who anticipated that strong and language-specific prior biases would prove necessary for acquisition, and those who anticipated that more general methods could suffice. The field of natural language processing has now developed methods that can measure the degree to which different kinds of input disambiguate different kinds of putative grammatical structure. For example, the work in this dissertation showed how variants of the DMV could be used to measure the evidential quality of word duration in conjunction with word terciles, and to compare a purely directional account (the Cond. model) with joint learning accounts.

Thus, computational methods are relevant to anybody with an interest in language

acquisition, not only those working in an empiricist, lexicalist, or otherwise statistically-focused theoretical paradigms, providing an opportunity for a convergence of empiricist/lexicalist and nativist/early-abstraction views. Just as computational models potentially show that some structures are readily identifiable without strong prior biases, they can also show that some specific prior bias identifies other structures, or identifies the same structures with much less data. Indeed, we saw in the Predictability DMV that a prior bias towards smaller parameterizations enabled effective learning, while the fully-joint model, that lacked backoff, did not. This is important evidence that should be of interest to language development researchers of all paradigms.

Next, we discuss potential elaborations to the computational model, potential laboratory experiments to test Predictability Bootstrapping, and finally conclude the dissertation.

7.3 Future work: modeling

The Predictability DMV is a model of syntax and word duration, and so obvious improvements include better models of syntax, and better models of word duration. The most obvious elaboration would move to a continuous measure of word duration. This could be achieved in the fully-generative model by incorporating a mixture of gaussians over word durations. The full model would then be similar to the current model, but would be free to adjust the boundaries of the terciles depending on such factors as head word identity and arc direction. Such a model would thus explicitly model reduction as a latent variable, with observed duration as its reflex, and potentially facilitate better learning.

The syntactic model could be improved by adopting Blunsom and Cohn's 2010 approach, who used a hierarchical non-parametric Pitman-Yor process to define a tree substitution grammar (TSG) factorization of dependency trees, whose factors may include multiple arcs. They use the same kind of backoff technology to embed Headden et al.'s (2009) model into their own, and achieved state-of-the-art results on the Wall Street Journal corpus. A TSG-based representation could enable the model to discover more effects of syntactic predictability. The current model distinguishes only first and non-first arcs, for example, which complicates the prospect of learning about ditransitive constructions as distinct from transitive constructions with a rightward adjunct. A TSG could potentially learn a "ditransitive" tree fragment with two rightward arcs that specializes in generating those dependents that are likely in a ditransitive construction,

while building adjuncts out of the individual arcs. In the first work on syntactic predictability effects, Gahl and Garnsey (2004) looked at verbs that were biased towards one version of the dative alternation, and a syntactic model that explicitly represented the dative alternation in this way could potentially exploit effects of such a dative bias.

Another potential improvement to the syntactic model could involve generalizations over words. The current model contains no notion of morphology or part-of-speech (except for the words and POS tag models that were given hand-annotated POS tags). As a first step, we could impose two levels of backoff, conditioning on words, words plus word duration, and words plus word duration and latent POS tag. Most POS induction systems note that POS tags can largely be described in terms of linear dependencies: nouns follow determiners and adjectives, determiners follow the beginning of the utterance and prepositions, and so on (Merialdo, 1994; Banko and Moore, 2004; Wang and Schuurmans, 2005; Goldwater and Griffiths, 2007). If we draw the factor graph assumed by the DMV on a sequence of words, together with the factor graph assumed by a sequence model, the resulting graphical model contains loops that complicate exact inference. However, Auli and Lopez (2011), in the context of learning a Combinatorial Categorial Grammar model with supertags, showed how loopy belief propagation could be used to simultaneously learn approximate parameters for a hierarchical model and a sequence model. This technology could be adapted to learn a DMV-based posterior together with a traditional sequence-based model of POS tags.

Other POS induction systems rely on situating word types in a vector space, and clustering word types into latent POS tag classes (Schütze, 1995; Christodoulopoulos et al., 2011; Lamar et al., 2010; Toutanova and Johnson, 2007). The vector space can be parameterized in terms of the local context of word types and so incorporate similar kinds of local sequence information as the explicit sequence models. However, the vector space can also be parameterized in terms of other features, including aspects of word form that may permit a latent morphological analysis. A clustering approach to POS tagging that did not invoke sequential dependencies could thus be incorporated into the current model and allow generalization across word types.

Additionally, the models that we used were well-suited for discovering the relevant correlations at a computational level, but it is possible that children not only look for correlations between syntactic probability and word reduction but enforce such a correlation. This is a kind of advanced knowledge about the form of a model called *regularization*. In the case of Predictability Bootstrapping, we would want to enforce a positive, linear correlation between the posterior generative entropies of syntactic

events and the duration of their head words, which could be implemented through *posterior regularization* (Graça et al., 2009; Gillenwater et al., 2011). Posterior regularization is typically used to enforce a *posterior sparsity* constraint. In dependency parsing, this means constraining the posterior so that each word is generated by only one or two heads with high probability, leading to a less ambiguous posterior over full trees.[1] However, posterior regularization could easily be used to enforce a correlation between posterior generative entropy and head word duration. Moreover, since Variational Bayes involves a modification to the M-step of EM, but posterior regularization involves a modification to the E-step, it should be possible to enforce the posterior correlation by simply modifying the E-step while still incorporating our prior biases to back off in the M-step. The resulting system would find a posterior distribution that was both close to the true Bayesian posterior and enforced a correlation between generative entropy and head word duration.

By enforcing this correlation, rather than simply discovering it, the model could potentially take advantage of predictability effects for much less common words. The current model mostly ignores head word duration until we expect a particular word to be a head enough times to overwhelm the prior bias to backoff. This is good for a computational-level model that seeks only to measure and exploit correlations to the extent that they are evident in the data, but infants may rely on prior knowledge about predictability effects to take advantage of them before they are evident. A posterior-regularization model would allow us to quantify the benefit of such prior knowledge.

Finally, the models presented in this dissertation all relied on batch learning algorithms, which ignores the fact that children encounter evidence over time and incrementally improve their internal model. We could make our models incremental by using online variational bayes EM (Hoffman et al., 2010), which Kwiatkowski et al. (2012) used to train an incremental unsupervised CCG model of syntactic and semantic bootstrapping. However, because our batch-trained models indicate whether the evidence for particular relationships between word durations and syntax exist in the dataset, it should be noted that they do speak to the incremental case. Indeed, the online variational bayes EM essentially works by taking a kind of weighted average of several batches, where each batch can contain one utterance or many utterances. Incremental models are useful for studying the time course of language acquisition,

[1] Our examination of the generative entropies revealed a different kind of posterior sparsity, wherein each word's distribution over Dependents and Boundaries was relatively unambiguous; that is, $P(\text{Deps and Bounds}|\textit{head})$ was unambiguous, while posterior regularization allows us to also make $P(\textit{head}|\text{Dep})$ unambiguous.

and for assessing how much information must be stored about each observed instance. Batch models, however, are suitable for simply measuring and characterizing statistical dependencies themselves.

7.4 Future work: experiments

As mentioned, computational modeling is only a complement to laboratory experiments with real people, and so more work is needed to validate Predictability Bootstrapping as a likely strategy for infants. Here, we sketch potential experiments. First, while there is good evidence that children track statistical regularities in their input (e.g. Saffran et al., 1996), and good evidence that children attend to word duration cues (e.g. Seidl, 2007), it is not clear that children gather statistics over word durations in the way that would enable bootstrapping from predictability effects. Thus, it would be interesting to see if children will respond to differential transitional probabilities according to segment duration in an artificial language.

The most direct laboratory test of the Predictability Bootstrapping hypothesis would be analogous to the study of Morgan et al. (1987), who found that subjects learned an artificial language more readily when presented with input whose prosody impressionistically grouped syntactic constituents together. The Predictability Bootstrapping edition of this experiment would have some common constituents and some uncommmon constituents. Stimuli consistent with Predictability Bootstrapping would contain short pronunciations of head words of the common constituents and long pronunciations of head words of uncommon constituents, inverse-predictability stimuli would contain the opposite, stimuli consistent with Prosodic Bootstrapping would contain long pronunciations of phrase-final words, and neutral stimuli would have randomly assigned durations. Ideally, (versions of) this study would be run with both adult and infant subjects.

There are a number of potential patterns that might result from such an experiment. First, we may find that both adult and infant subjects learn the language best from the consistent stimuli, providing good evidence that Predictability Bootstrapping is an actual strategy among real language learners. Second, we may find that infants learn the language best from the predictability stimuli, but adults learn the language best from the prosodic stimuli. This result would indicate that the infants rely mostly on predictability effects, perhaps because they haven't learned a robust representation of prosodic phrasing, but adults, who do have robust prosodic representations, rely more

on prosodic cues. Additionally, this result would provide a partial explanation for the apparent "critical period" for language learning: children attend to relatively language-general cues about grammatical probability in suprasegmental variation, but adults attend to relatively language-specific cues about prosodic phrasing. Finally, we may find that infants perform equally well with both predictability-consistent and inverse-predictability stimuli, suggesting that the direction of the correlation between word duration and syntactic probability is not hard-coded (or at least easily over-ridden).

7.5 Conclusion

This dissertation sought to show that predictability effects may provide infants useful information about linguistic structure, and, in particular, that reduction in terms of word duration may provide important cues to syntactic structure. While the models developed to this end are certainly not complete, as demonstrated by the preceding discussion, they do provide good evidence that attending to word duration simplifies the acquisition task by providing cues to syntactic predictability and, to a lesser extent, prosodic phrasing. The work in this dissertation also makes a larger methodological point: when we hypothesize that the input available to children contains some kind of evidence about some linguistic structure, it is important to check the input to see how strong that evidence is. Specifically, the Prosodic Bootstrapping hypothesis proposed that prosodic boundaries cue syntactic boundaries by coinciding with them. While we did find that prosodic phrasing appeared to provide cues to syntactic phrasing, it seemed to do so by placing prosodic boundaries after syntactic boundaries, not coincident with syntactic boundaries. Computational modeling provides a powerful and necessary tool for measuring the quality of a purported source of evidence for a putative linguistic representation.

Bibliography

Abney, S. (1991). *Parsing by chunks*. Kluwer Academic Publishers, Dordrecht.

Abney, S. (1992). Prosodic structure, performance structure and phrase structure. In *Proceedings of Speech and Natural Language Workshop*, pages 425–428.

Anderson, A. H., Bader, M., Bard, E. G., Boyle, E., Doherty, G., Garrod, S., Isard, S., Kowtko, J., McAllister, J., Miller, J., Sotillo, C., and Thomspon, H. S. (1991). The HCRC map task corpus. *Language and Speech*, 34(4).

Auli, M. and Lopez, A. (2011). A comparison of loopy belief propagation and dual decomposition for integrated CCG supertagging and parsing. In *Proceedings of ACL*.

Aylett, M. and Turk, A. (2004). The smooth signal redundancy hypothesis: A functional explanation for relationships between redundancy, prosodic prominence, and duration in spontaneous speech. *Language and Speech*, 47(1):31–56.

Aylett, M. and Turk, A. (2006). Language redundancy predicts syllabic duration and the spectral characteristics of vocalic syllable nuclei. *Journal of the Acoustical Society of America*, 119(5):3048–3059.

Baker, J. (1979). Trainable grammars for speech recognition. In *Proceedings of the 97th meeting of the Acoustical Society of America*, pages 547–550, Cambridge, Mass.

Baker, M. C. (2002). *The atoms of language: The mind's hidden rules of grammar*. Basic Books.

Banko, M. and Moore, R. C. (2004). Part of speech tagging in context. In *Proceedings of COLING*.

Bannard, C., Lieven, E., and Tomasello, M. (2009). Modeling children's early grammatical knowledge. *PNAS*, 106(41):17284–9.

Bard, E. G., Anderson, A. H., Sotillo, C., Aylett, M., Doherty-Sneddon, G., and Newlands, A. (2000). Controlling the intelligibility of referring expressions in dialogue. *Journal of Memory and Language*, 42:1–22.

Barr, D. J., Levy, R., Scheepers, C., and Tily, H. (2012). Random effects structure in mixed-effect models: Keep it maximal. *Journal of Memory and Language*.

Beckman, M., Hirschberg, J., and Shattuck-Hufnagel, S. (2005). The original ToBI system and the evolution of the ToBI framework. In Jun, S.-A., editor, *Prosodic Typology–The Phonology of Intonation and Phrasing*. Oxford University Press.

Beckman, M. and Pierrehumbert, J. (1986). Intonational structure in Japanese and English. *Phonology Yearbook*, 3:255–309.

Beckman, M. E. and Edwards, J. (1990). Lengthenings and shortenings and the nature of prosodic constituency. In Kingston, J. and Beckman, M. E., editors, *Between the grammar and physics of speech: Papers in laboratory phonology I*, pages 152–178. Cambridge: Cambridge University Press.

Bell, A., Brenier, J. M., Gregory, M., Girand, C., and Jurafsky, D. (2009). Predictability effects on durations of content and function words in conversational English. *Journal of Memory and Language*, 60:92–111.

Bliss, T. V. P. and Gardner-Medwin, T. (1973). Long-lasting potentiation of synaptic transmissions in the dendate area of unanaesthetized rabbit following stimulation of the perforant path. *Journal of Physiology*, 232:357–374.

Bliss, T. V. P. and Lømo, T. (1973). Long-lasting potentiation of synaptic transmissions in the dendate area of anaesthetized rabbit following stimulation of the perforant path. *Journal of Physiology*, 232:357–374.

Blunsom, P. and Cohn, T. (2010). Unsupervised induction of tree substitution grammars for dependency parsing. In *Proceedings of EMNLP*, pages 1204 – 1213.

Boyle, E. A., Anderson, A. H., and Newlands, A. (1994). The effects of visibility on dialogue and performance in a cooperative problem solving task. *Language and Speech*, 70(1).

Brent, M. R. and Siskind, J. M. (2001). The role of exposure to isolated words in early vocabulary development. *Cognition*, 81:31–44.

Bresnan, J., Carletta, J., Crouch, R., Nissim, M., Steedman, M., Wasow, T., and Zaenen, A. (2002). Paraphrase analysis for improved generation. In *LINK project: HRCR Edinburgh-CLSI Stanford*.

Bruner, J. S. (1975). The ontogenesis of speech acts. *Journal of Child Language*, 2:1–19.

Calhoun, S., Carletta, J., Brenier, J., Mayo, N., Jurafsky, D., Steedman, M., and Beaver, D. (2010). The NXT-format Switchboard corpus: A rich resource for investigating the syntax, semantics, pragmatics and prosody of dialogue. *Language Resources and Evaluation*, 44(4):387–419.

Carroll, G. and Charniak, E. (1992). Two experiments on learning probabilistic dependency grammars from corpora. Technical Report CS-92-16, Brown University, Providence, RI, USA.

Chang, F., Dell, G. S., and Bock, K. (2006a). Becoming syntactic. *Psychological Review*, 113(2):234–272.

Chang, F., Lieven, E., and Tomasello, M. (2006b). Using child utterances to evaluate syntax acquisition algorithms. In *Proceedings of the Cognitive Science Society*, pages 154 – 159.

Chen, S. F. and Goodman, J. T. (1996). An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 310–318.

Chomsky, N. (1980). *Rules and Representations*. Basil Blackwell, Oxford.

Christodoulopoulos, C., Goldwater, S., and Steedman, M. (2011). A bayesian mixture model for PoS induction using multiple features. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 638–647, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Coco, M. I. and Keller, F. (2010). Scan pattern in visual scenes predict sentence production. In *Proceedings of the 32nd Annual Conference of the Cognitive Science Society*.

Cohen, S. B. and Smith, N. A. (2008). Logistic normal priors for unsupervised probabilistic grammar induction. In *Advances in Neural Information Processing Systems 22*.

Dahan, D., Magnuson, J. S., and Tanenhaus, M. K. (2001). Time course of frequency effects in spoken-word recognition: evidence from eye movements. *Cognitive Psychology*, 42:317–367.

de Marneffe, M.-C. and Manning, C. D. (2008). Stanford typed dependencies manual. Technical report, Stanford University.

Dell, G. S. (1986). A spreading-activation theory of retrieval in sentence production. *Psychological Review*, 93(3):283–321.

Dreyer, M. and Shafran, I. (2007). Exploiting prosody for pcfgs with latent annotations. In *Proceedings of Interspeech*, Antwerp, Belgium.

Ferreira, V. S. and Dell, G. S. (2000). The effect of ambiguity and lexical availability on syntactic and lexical production. *Cognitive Psychology*, 40:296–340.

Fisher, C. (1996). Structural limits on verb mapping: The role of analogy in childrens' interpretations of sentences. *Cognitive Psychology*, 31:41–81.

Fisher, C. (2002). The role of abstract syntactic knowledge in language acquisition: A reply to Tomasello (2000). *Cognition*, 82(3):259–278.

Fisher, C., Klingler, S. L., and Song, H. (2006). What does syntax say about space? 2-year-olds use sentence structure to learn new prepositions. *Cognition*, 101:B19–B29.

Frank, A. and Jaeger, T. F. (2008). Speaking rationally: Uniform information density as an optimal strategy for language production. In *Proceedings of CogSci*, pages 933–938.

Freudenthal, D., Pinea, J. M., Aguado-Orea, J., and Gobet, F. (2007). Modeling the developmental patterning of finiteness marking in English, Dutch, German, and Spanish using MOSAIC. *Cognitive Science*, 31:311–341.

Freudenthal, D., Pinea, J. M., and Gobet, F. (2006). Modeling the development of children's use of optional infinitives in Dutch and English using MOSAIC. *Cognitive Science*, 30:277–310.

Gahl, S. and Garnsey, S. M. (2004). Knowledge of grammar, knowledge of usage: Syntactic probabilities affect pronunciation variation. *Language*, 80:748–775.

Gahl, S., Garnsey, S. M., Fisher, C., and Matzen, L. (2006). "That sounds unlikely": Syntactic probabilities affect pronunciation. In *Proceedings of the 27th meeting of the Cognitive Science Society*.

Gee, J. P. and Grosjean, F. (1983). Performance structures: A psycholinguistic and linguistic appraisal. *Cognitive Psychology*, 15:411–458.

Gillenwater, J., Ganchev, K., Graça, J., Pereira, F., and Taskar, B. (2011). Posterior sparsity in unsupervised dependency parsing. *Journal of Machine Learning Research*, 12:455–490.

Gimpel, K. and Smith, N. A. (2012). Concavity and initialization for unsupervised dependency parsing. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 577–581, Montréal, Canada. Association for Computational Linguistics.

Gleitman, L. (1990). The structural sources of verb meanings. *Language Acquisition*, 1(1):3–55.

Gleitman, L. and Wanner, E. (1982). Language acquisition: The state of the art. In Wanner, E. and Gleitman, L., editors, *Language acquisition: The state of the art*, pages 3–48. Cambridge University Press, Cambridge, UK.

Gold, E. M. (1967). Language identification in the limit. *Information and Control*, 10(5):447–474.

Goldwater, S. and Griffiths, T. L. (2007). A fully bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic.

Goldwater, S., Griffiths, T. L., and Johnson, M. (2009). A bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, 112(1):21–54.

Graça, J., Ganchev, K., Taskar, B., and Pereira, F. (2009). Posterior vs. parameter sparsity in latent variable models. In *Proceedings of NIPS*.

Gregory, M. L., Johnson, M., and Charniak, E. (2004). Sentence-internal prosody does not help parsing the way punctuation does. In *Proceedings of the North American Association for Computational Linguistics (NAACL)*, pages 81–88.

Griffin, Z. M. and Bock, K. (1998). Constraint, word frequency, and the relationship between lexical processing levels in spoken word production. *Journal of Memory and Language*, 38:313–338.

Headden, W., Johnson, M., and McClosky, D. (2009). Improved unsupervised dependency parsing with richer contexts and smoothing. In *NAACL-HLT*.

Hoffman, M., Blei, D. M., and Bach, F. (2010). Online learning for latent dirichlet allocation. In *Proceedings of NIPS*.

Jaeger, T. F. (2010). Redundancy and reduction: Speakers manage syntactic information density. *Cognitive Psychology*, 61:23–62.

Johansson, R. and Nugues, P. (2007). Extended constituent-to-dependency conversion for English. In *Proceedings of NODALIDA 2007*.

Johnson, M., Jones, B., Demuth, K., and Frank, M. C. (2010). Synergies in learning words and their referents. *Neural Information Processing Systems*, 23.

Kahn, J. G., Lease, M., Charniak, E., Johnson, M., and Ostendorf, M. (2005). Effective use of prosody in parsing conversational speech. In *Proceedings of HLT-EMNLP*, pages 233–240.

Klatt, D. H. (1976). Linguistic uses of segmental durations in English: Acoustic and perceptual evidence. *JASA*, 59:1208–1221.

Klein, D. and Manning, C. D. (2004). Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of ACL*, pages 479–486.

Kromann, M. T. (2003). The Danish Dependency Treebank and the DTAG treebank tool. In *Proceedings of the Second Workshop on Treebanks and Linguistic Theories*, pages 217–220.

Kurihara, K. and Sato, T. (2006). Variational Bayesian grammar induction for natural language. In *International Colloquium on Grammatical Inference*, pages 84–96.

Kwiatkowski, T., Goldwater, S., Zettlemoyer, L., and Steedman, M. (2012). A probabilistic model of syntactic and semantic acquisition from child-directed utterances and their meanings. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Ladd, B. (1996). *Intonational Phonology*. Cambridge University Press.

Lamar, M., Maron, Y., Johnson, M., and Bienenstock, E. (2010). SVD and clustering for unsupervised POS tagging. In *Proceedings of ACL 2010 Conference Short Papers*, pages 215–219.

Levin, B. (1993). *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press, Chicago, IL.

Liberman, M. and Prince, A. (1977). On stress and linguistic rhythm. *Linguistic Inquiry*, 8(2):249–336.

Lindblom, B. (1990). *Explaining phonetic variation: A sketch of the H & H theory*, pages 403–439. Kluwer Academic Publishers.

Mackay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press. http://www.inference.phy.cam.ac.uk/mackay/itila/.

MacWhinney, B. (2000). *The CHILDES project: Tools for analyzing talk*. Lawrence Erlbaum Associates, Mahwah, NJ, third edition.

Marr, D. (1982). *Vision: A Computational approach*. Freeman & Co., San Francisco.

McCallum, A. K. (2002). Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

McClelland, J. L. and Elman, J. (1986). The TRACE model of speech recognition. *Cognitive Psychology*, 18:1–86.

Mehler, J., Juszcyk, P., Lambertz, G., Halsted, N., Bertoncini, J., and Amiele-Tison, C. (1988). A precursor to language acquisition in young infants. *Cognition*, 29:143–178.

Mel'čuk, I. (1988). *Dependency Syntax: theory and practice*. SUNY Press, Albany, NY.

Merialdo, B. (1994). Tagging english text with a probabilistic model. *Computational Linguistics*, 20(2):155–172.

Millotte, S., Wales, R., and Christophe, A. (2007). Phrasal prosody disambiguates syntax. *Language and Cognitive Processes*, 22(6):898–909.

Molina, A. and Pla, F. (2002). Shallow parsing using specialized HMMs. *Journal of Machine Learning Research*, 2:595–613.

Morgan, J. L. and Demuth, K. (1996). *Signal to Syntax: An overview*. Psychology Press, Erlbaum, NJ.

Morgan, J. L., Meier, R. P., and Newport, E. L. (1987). Structural packaging in the input to language learning: contributions of prosodic and morphological marking of phrases to the acquisition of language. *Cognitive Psychology*, 19:498–550.

Naigles, L. (1990). Children use syntax to learn verb meanings. *Journal of Child Language*, 17:357–374.

Nazzi, T., Nelson, D. G. K., Jusczyk, P. W., and Jusczyk, A. M. (2000). Six-month-olds' detection of clauses embedded in continuous speech: effects of prosodic well-formedness. *Infancy*, 1:123–147.

Nefian, A. V., Liang, L., Pi, X., Xiaoxiang, L., Moe, C., and Murphy, K. (2002). A coupled HMM for audiovisual speech recognition. In *International Conference on Acoustics, Speech and Signal Processing*.

Nespor, M. and Vogel, I. (1986). *Prosodic Phonology*. Foris Publications.

Nivre, J., Hall, J., Kübler, S., McDonald, R., Nilsson, J., Riedel, S., and Yuret, D. (2007). The CoNLL 2007 shared task on dependency parsing. In *Proceedings of CoNLL-2007*.

Nöth, E., Batliner, A., Kieling, A., and Kompe, R. (2000). Verbmobil: The use of prosody in the linguistic components of a speech understanding system. *IEEE Transactions on Speech and Audio Processing*, 8(5).

Osborne, M. (2000). Shallow parsing as part-of-speech tagging. In *Proceedings of CoNLL-2000 and LLL-2000*, pages 145 – 147.

Paskin, M. A. (2001). Grammatical bigrams. In Becker, T. D. S. and Gharahmani, Z., editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA. MIT Press.

Pate, J. K. and Goldwater, S. (2011a). Predictability effects in infant-directed and adult-directed speech: Does the listener matter? In *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*.

Pate, J. K. and Goldwater, S. (2011b). Unsupervised syntactic chunking with acoustic cues: computational models for prosodic bootstrapping. In *Proceedings of the 2nd ACL workshop on Cognitive Modeling and Computational Linguistics*.

Pate, J. K. and Goldwater, S. (2013). Unsupervised dependency parsing with acoustic cues. *Transactions of the ACL*.

Pereira, F. and Schabes, Y. (1992). Inside-outside reestimation from partially bracketed corpora. In *Proceedings of the 30th Annual Meeting of the ACL*, pages 128–135.

Pinker, S. (1984). *Language Learnability and Language Development*. Harvard University Press, Cambridge, MA.

Pinker, S. (1994). *The Language Instinct*. Harper Perennial Modern Classics, New York.

Pinker, S., Lebeaux, D. S., and Frost, L. A. (1987). Productivity and constraints in the acquisition of the passive. *Cognition*, 26:195–267.

Pluymaekersa, M., Ernestusb, M., and Baayen, R. H. (2005). Articulatory planning is continuous and sensitive to informational redundancy. *Phonetica*, 62:146–159.

Ponvert, E., Baldridge, J., and Erk, K. (2010). Simple unsupervised identification of low-level constituents. In *ICSC*.

Ponvert, E., Baldridge, J., and Erk, K. (2011). Simple unsupervised grammar induction from raw text with cascaded finite state models. In *Proceedings of ACL-HLT*.

Price, P. J., Ostendorf, M., Shattuck-Hufnagel, S., and Fong, C. (1991). The use of prosody in syntactic disambiguation. *JASA*, pages 2956–2970.

Priva, U. C. (2008). Using information content to predict phone deletion. In *Proceedings of the 27th west coast conference on formal linguistics*, pages 90–98.

Ratnaparkhi, A. (1996). A maximum entropy part-of-speech tagger. In *Proceedings of EMNLP*, pages 82 – 94.

Rytting, C. A., Brew, C., and Fosler-Lussier, E. (2010). Segmenting words from natural speech: subsegmental variation in segmental cues. *Journal of Child Language*, 37(3):513–543.

Saffran, J., Newport, E. L., and Aslin, R. N. (1996). Word segmentation: the role of distributional cues. *Journal of Memory and Language*, 35:606–621.

Schütze, H. (1995). Distributional part-of-speech tagging. In *Proceedings of EACL*, pages 141–148.

Schwartz, R., Abend, O., Reichart, R., and Rappoport1, A. (2011). Neutralizing linguistically problematic annotations in unsupervised dependency parsing evaluation. In *Proceedings of the 49th ACL*, pages 663–672.

Seginer, Y. (2007). Fast unsupervised incremental parsing. In *Proceedings of the Association of Computational Linguistics*.

Seidl, A. (2007). Infants' use and weighting of prosodic cues in clause segmentation. *Journal of Memory and Language*, 57(1):24–48.

Selkirk, E. O. (1978). *On prosodic structure and its relation to syntactic structure*. TAPIR, Trondheim.

Selkirk, E. O. (1984). *Phonology and Syntax*. MIT Press, Cambridge, Mass.

Sha, F. and Pereira, F. (2003). Shallow parsing with conditional random fields. In *Proceedings of HLT-NAACL 03*, pages 213–220.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423.

Shattuck-Hufnagel, S. and Turk, A. E. (1996). A prosody tutorial for investigators of auditory sentence processing. *Journal of Psycholinguistic Research*, 25(2):193–247.

Smith, N. A. and Eisner, J. (2005). Guiding unsupervised grammar induction using contrastive estimation. In *International Joint Conference on Artificial Intelligence (IJCAI) Workshop on Grammatical Inference Applications*, pages 73–82, Edinburgh.

Soderstrom, M., Seidl, A., Nelson, D. G. K., and Jusczyk, P. W. (2003). The prosodic bootstrapping of phrases: Evidence from prelinguistic infants. *Journal of Memory and Language*, 49:249–267.

Spitkovsky, V., Alshawi, H., and Jurafsky, D. (2010). From baby steps to leapfrog: How "less is more" in unsupervised dependency parsing. In *NAACL-HLT*.

Steedman, M. (1996). *Phrasal Intonation and the Acquisition of Syntax*. Psychology Press, Erlbaum, NJ.

Sutton, C. (2006). Grmm: Graphical models in mallet. http://mallet.cs.umass.edu/grmm/.

Swingley, D. and Aslin, R. N. (2007). Lexical competition in young childrens' word learning. *Cognitive Psychology*, 54:99–132.

Tily, H., Gahl, S., Arnon, I., Snider, N., Kothari, A., and Bresnan, J. (2009). Syntactic probabilities affect pronunciation variation in spontaneous speech. *Language and Cognition*, 1(2):147–165.

Tjong, E. F., Sang, K., and Buchholz, S. (2000). Introduction to the CoNLL-2000 shared task: Chunking. In *Proceedings of CoNLL-2000 and LLL-2000*, Lisbon, Portugal.

Tomasello, M. (2000). Do young children have adult syntactic competence? *Cognition*, 74:209–253.

Toutanova, K. and Johnson, M. (2007). A bayesian lda-based model for semi-supervised part-of-speech tagging. In *Proceedings of NIPS*.

Turk, A. (2010). Does prosodic constituency signal relative predictability? a smooth signal redundancy hypothesis. In *Proceedings of Labphon 2010*.

Wagner, L. (2010). Inferring meaning from syntactic structures in acquisition: The case of transitivity and telicity. *Language and Cognitive Processes*, 25:1354–1379.

Wang, Q. I. and Schuurmans, D. (2005). Improved estimation for unsupervised part-of-speech tagging. In *Proceedings of the IEEE International Conference on Natural Language Processing and Knowledge Engineering (IEEE NLP-KE)*.

Wexler, K. and Culicover, P. W. (1983). *Formal principles of language acquisition*. MIT Press, Cambridge, MA.

Wightman, C. W., Shattuck-Hufnagel, S., Ostendorf, M., and Price, P. J. (1992). Segmental durations in the vicinity of prosodic phrase boundaries. *Journal of the Acoustical Society of America*, 91(3):1707–1717.

Yuret, D. (1998). Lexical attraction models of language.

Zhao, Y. and Jurafsky, D. (2009). The effect of lexical frequency and Lombard reflex on tone hyperarticulation. *Journal of Phonetics*, 37:231–247.

Zipf, G. K. (1949). *Human behavior and the principle of least effort*. Addison-Wesley.